Introduction to Internet Applications

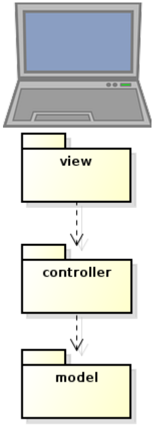
Internet Applications, ID1354

Contents

Section

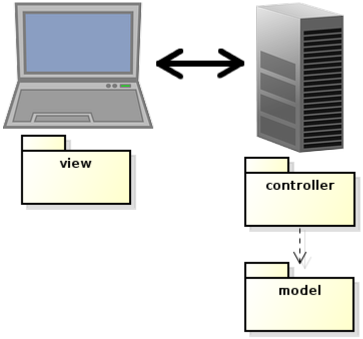
- Distributed Architectures
- HTTP and Other Protocols
- Tools
- User Interface Design

Local Application



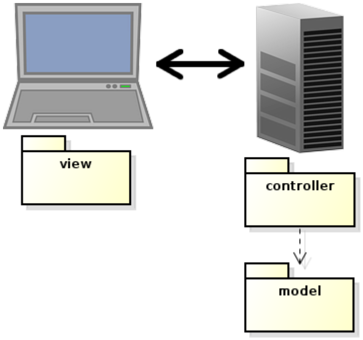
- We are familiar with an architecture where the entire application resides on the same computer.

Introducing a Server



- Now, the application will be split on two tiers (computers).

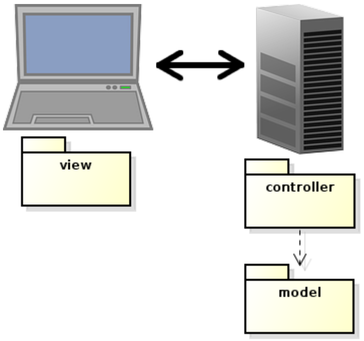
Introducing a Server



- Now, the application will be split on two tiers (computers).
- A client that has the view and a server that has controller and model.

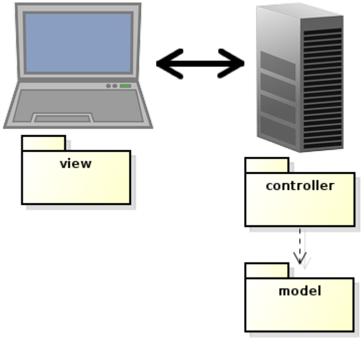
.

Introducing a Server



- Now, the application will be split on two tiers (computers).
- A client that has the view and a server that has controller and model.
- The view is displayed in a web browser.

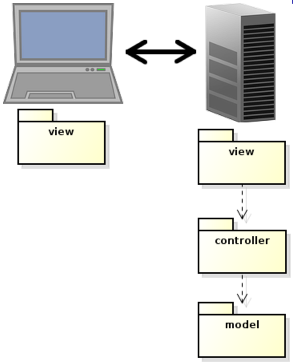
Introducing a Server



- Now, the application will be split on two tiers (computers).
- A client that has the view and a server that has controller and model.
- The view is displayed in a web browser.

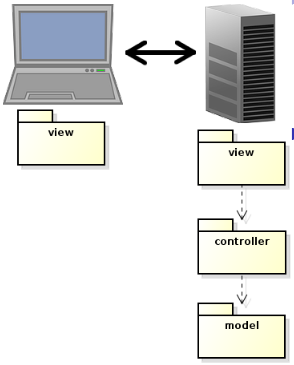
This architecture is not good, we also need layers for communication.

Server-Side Communication



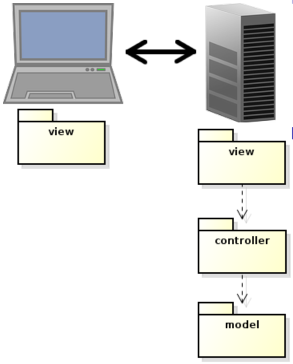
- First, we add a server layer, normally called view (a bit confusing).

Server-Side Communication



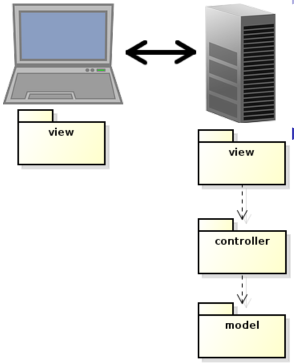
- First, we add a server layer, normally called view (a bit confusing).
- However, the server side view layer performs tasks typical of a view:

Server-Side Communication



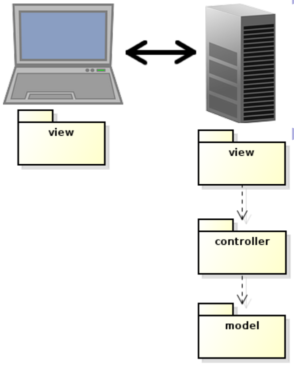
- First, we add a server layer, normally called view (a bit confusing).
- However, the server side view layer performs tasks typical of a view:
  - Creates views (HTML), which are sent to the client.

Server-Side Communication



- First, we add a server layer, normally called view (a bit confusing).
- However, the server side view layer performs tasks typical of a view:
  - Creates views (HTML), which are sent to the client.
  - Interprets user gestures, a click in a web page creates a request to the server.

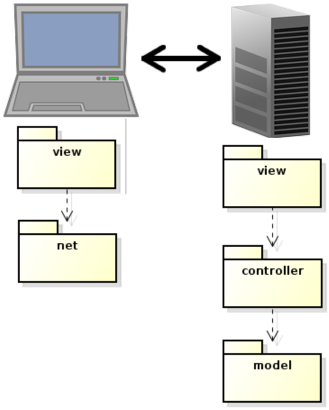
Server-Side Communication



- First, we add a server layer, normally called view (a bit confusing).
- However, the server side view layer performs tasks typical of a view:
  - Creates views (HTML), which are sent to the client.
  - Interprets user gestures, a click in a web page creates a request to the server.

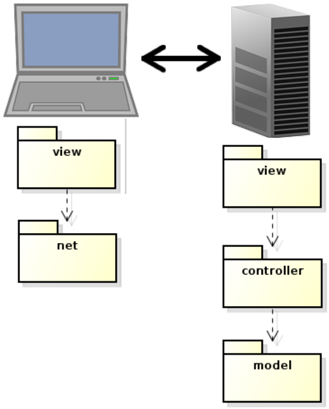
It might seem that we need yet a layer, for network handling. There is such a layer, but it is in the web server. We don't write it ourselves.

Client-Side Communication



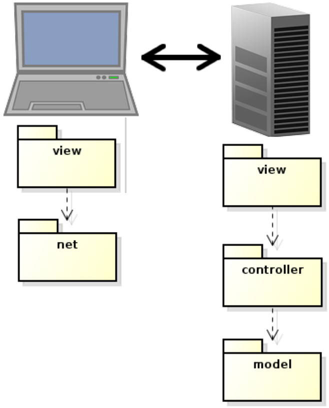
- Next, we add a client layer for communication, the net layer.

Client-Side Communication



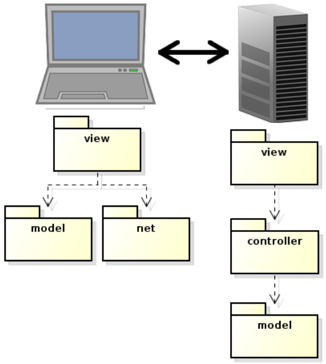
- Next, we add a client layer for communication, the net layer.
- Actually, the browser handles most of the communication.
  - The small network code written by us is normally considered part of the client-side view, the net layer is omitted.

Client-Side Communication



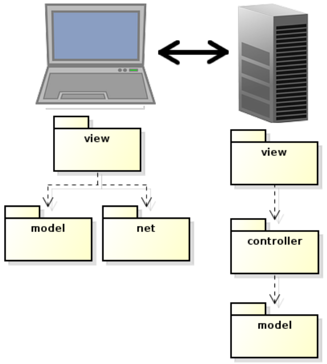
- Next, we add a client layer for communication, the net layer.
- Actually, the browser handles most of the communication.
    - The small network code written by us is normally considered part of the client-side view, the net layer is omitted.
- This is a traditional web application.

The MVVM Pattern



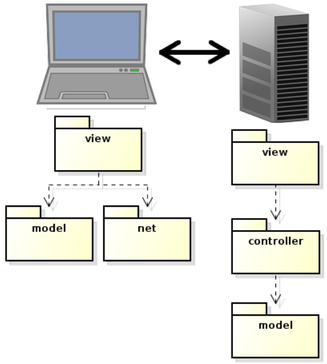
- The trend is that data is stored also on the client, therefore we get a client-side model.

The MVVM Pattern



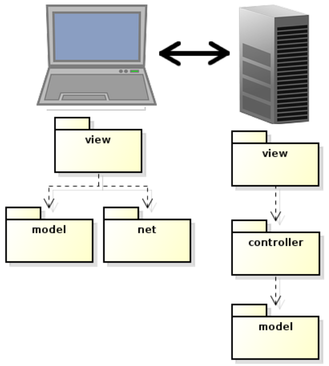
- The trend is that data is stored also on the client, therefore we get a client-side model.
- This reduces the network communication, since we do not need to resend the entire view each time the user does something.

The MVVM Pattern



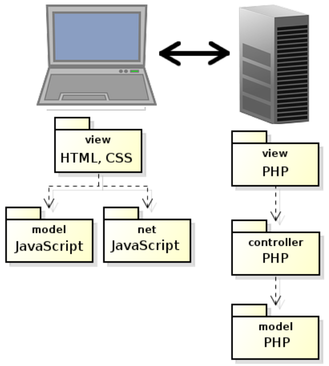
- The trend is that data is stored also on the client, therefore we get a client-side model.
- This reduces the network communication, since we do not need to resend the entire view each time the user does something.
- Thereby, the application becomes faster.

The MVVM Pattern



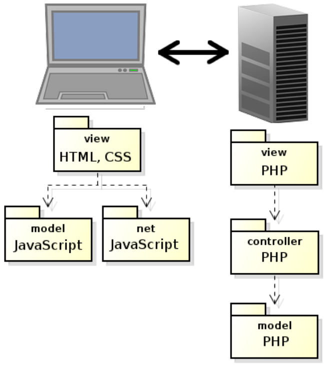
- The trend is that data is stored also on the client, therefore we get a client-side model.
- This reduces the network communication, since we do not need to resend the entire view each time the user does something.
- Thereby, the application becomes faster.
- This is referred to as the MVVM, model-view-viewmodel pattern.

Programming Languages



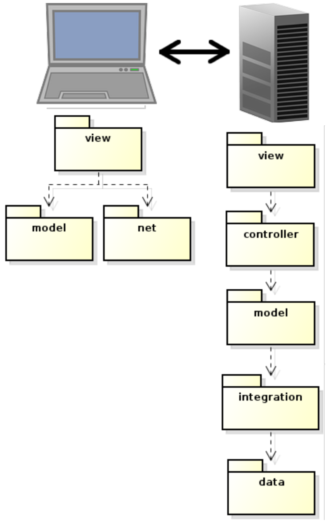
- This is the architecture we will normally use during the course.

Programming Languages



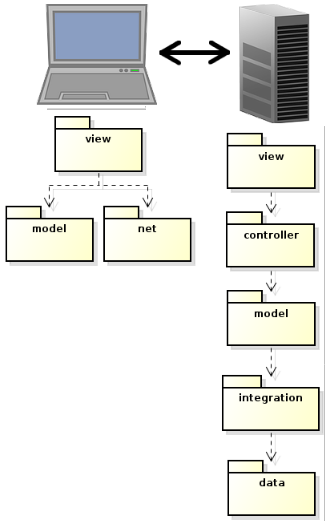
- This is the architecture we will normally use during the course.
- The view is programmed in HTML and CSS, client side behavior is programmed in JavaScript and the entire server side code is written in PHP.

Three-Tier Architecture



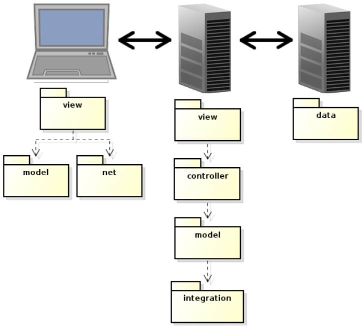
- Of course, we also need to store data. That is done in the data layer, which is often a database.

Three-Tier Architecture



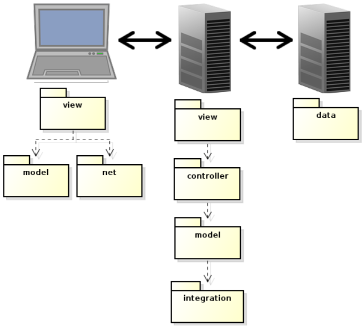
- Of course, we also need to store data. That is done in the data layer, which is often a database.
- We also introduce the integration layer, to handle the database calls.

Three-Tier Architecture (Cont'd)



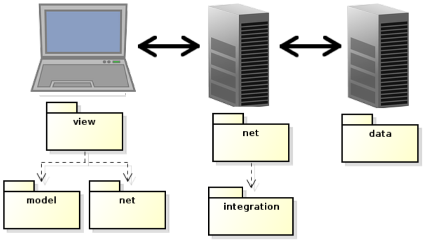
- In a bigger application, we would most likely place the database in a separate node.

Three-Tier Architecture (Cont'd)



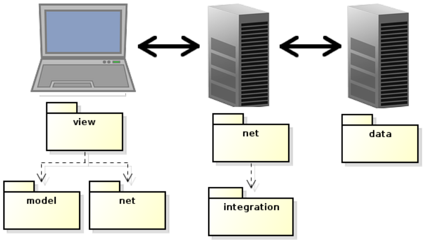
- In a bigger application, we would most likely place the database in a separate node.
- This is called three-tier architecture and is, since long time, the dominating architecture for web applications.

Event-Driven Architecture



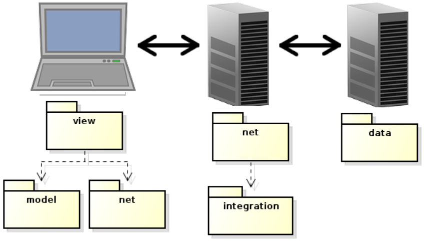
- In the latest year, there is a growing tendency to move business logic to the client, perhaps completely remove the server-side model.

Event-Driven Architecture



- In the latest year, there is a growing tendency to move business logic to the client, perhaps completely remove the server-side model.
- This is made possible with web sockets, which enable full duplex browser-server communication.

Event-Driven Architecture



- In the latest year, there is a growing tendency to move business logic to the client, perhaps completely remove the server-side model.
- This is made possible with web sockets, which enable full duplex browser-server communication.
- The motive is to reduce communication latency. The browser informs the server about user actions, but does not wait for response before updating the view.

Section

- Distributed Architectures
- HTTP and Other Protocols
- Tools
- User Interface Design

The IP Protocol

- All Internet communication is based on the Internet Protocol (IP).

The IP Protocol

- All Internet communication is based on the Internet Protocol (IP).
- IP provides basic functionality for sending and receiving data.

The IP Protocol

- All Internet communication is based on the Internet Protocol (IP).
- IP provides basic functionality for sending and receiving data.
- Data is sent in chunks, called packages. A package is like an envelope for a letter. It has sender and a receiver addresses and a content, which is the data being transmitted.

The IP Protocol

- All Internet communication is based on the Internet Protocol (IP).
- IP provides basic functionality for sending and receiving data.
- Data is sent in chunks, called packages. A package is like an envelope for a letter. It has sender and a receiver addresses and a content, which is the data being transmitted.
- A node (computer) receiving a packet can accept it, ignore it or retransmit it.

The IP Protocol

- All Internet communication is based on the Internet Protocol (IP).
- IP provides basic functionality for sending and receiving data.
- Data is sent in chunks, called packages. A package is like an envelope for a letter. It has sender and a receiver addresses and a content, which is the data being transmitted.
- A node (computer) receiving a packet can accept it, ignore it or retransmit it.
- A node dedicated to retransmitting packets across subnet borders is called a router.

IP Address

- An internet (version 4) address has 32 bits divided into four bytes, [0-255].[0-255].[0-255].[0-255]. Each node connected to the internet has one or more addresses.

IP Address

- An internet (version 4) address has 32 bits divided into four bytes, [0-255].[0-255].[0-255].[0-255]. Each node connected to the internet has one or more addresses.
- Normally, an IP address must be unique, assigned only to one node.

IP Address

- An internet (version 4) address has 32 bits divided into four bytes, [0-255].[0-255].[0-255].[0-255]. Each node connected to the internet has one or more addresses.
- Normally, an IP address must be unique, assigned only to one node.
- Some addresses, like 192.168.X.X are dedicated to private networks and can be used freely. Such an address is not transmitted on the public internet. Instead, it is translated to a public address by a router.

The TCP Protocol

- TCP, Transmission Control Protocol, is used on top of the IP protocol.

The TCP Protocol

- TCP, Transmission Control Protocol, is used on top of the IP protocol.
- TCP adds transport guarantees, for example the following.

The TCP Protocol

- TCP, Transmission Control Protocol, is used on top of the IP protocol.
- TCP adds transport guarantees, for example the following.
    - Packets are delivered to the receiver in the same order they are sent by the sender.

The TCP Protocol

- TCP, Transmission Control Protocol, is used on top of the IP protocol.
- TCP adds transport guarantees, for example the following.
  - Packets are delivered to the receiver in the same order they are sent by the sender.
  - Delivered packets have the same content as sent packets.

The TCP Protocol

- TCP, Transmission Control Protocol, is used on top of the IP protocol.
- TCP adds transport guarantees, for example the following.
  - Packets are delivered to the receiver in the same order they are sent by the sender.
  - Delivered packets have the same content as sent packets.
  - There are no lost packets.

The TCP Protocol (Cont'd)

- TCP is connection-oriented, think of a telephone line as opposed to sending a letter. To establish a TCP connection is a slow operation.

The TCP Protocol (Cont’d)

- TCP is connection-oriented, think of a telephone line as opposed to sending a letter. To establish a TCP connection is a slow operation.
- TCP handles ports, which makes it possible to have multiple connections with the same IP address open simultaneously. A port is identified by a number. An endpoint of a TCP connection has an IP address and a port number.

DNS

- IP addresses are normally translated to names (instead of numbers). Such a name is called domain name.

DNS

- IP addresses are normally translated to names (instead of numbers). Such a name is called domain name.
- Domain names are divided into subdomains, divided by dots (`.`)
  - The address **`www.ict.kth.se`** consists of the subdomain **`www`**, which is part of the subdomain **`ict`**, which is part of **`kth`**, which is part of **`se`**, which is part of the root, **`.`**

DNS

- IP addresses are normally translated to names (instead of numbers). Such a name is called domain name.
- Domain names are divided into subdomains, divided by dots (`.`)
  - The address **`www.ict.kth.se`** consists of the subdomain **`www`**, which is part of the subdomain **`ict`**, which is part of **`kth`**, which is part of **`se`**, which is part of the root, `.`
- The translation between numbers and names is managed by DNS, Domain Name System.

URL

- A Uniform Resource Locator, URL defines a resource’s location on the internet.

URL

- A Uniform Resource Locator, URL defines a resource's location on the internet.
- A URL consists of four parts.
  1. A protocol, e.g., **`http`**

URL

- A Uniform Resource Locator, URL defines a resource's location on the internet.
- A URL consists of four parts.
  1. A protocol, e.g., **`http`**
  2. A host (IP address or name),
     `http://www.kth.se`

URL

- A Uniform Resource Locator, URL defines a resource's location on the internet.
- A URL consists of four parts.
  1. A protocol, e.g., **`http`**
  2. A host (IP address or name), **`http://www.kth.se`**
  3. A port number (optional). The default HTTP port number is 80. **`http://www.kth.se:8080`**

URL

- A Uniform Resource Locator, URL defines a resource's location on the internet.
- A URL consists of four parts.
  1. A protocol, e.g., **`http`**
  2. A host (IP address or name), **`http://www.kth.se`**
  3. A port number (optional). The default HTTP port number is 80. **`http://www.kth.se:8080`**
  4. A path, which identifies the resource's location on the server. **`http://www.kth.se:8080/abc/index.html`**

URN and URI

- A Uniform Resource Name, URN is a resource identifier without host name and port number. A typical example is a isbn identifying a book.

URN and URI

- A Uniform Resource Name, URN is a resource identifier without host name and port number. A typical example is a isbn identifying a book.
- A Uniform Resource Identifier, URI is either a URL or URN.

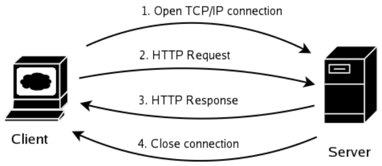
HTTP

- HyperText Transfer Protocol, HTTP is used for communication between web browsers and web servers.

HTTP

- HyperText Transfer Protocol, HTTP is used for communication between web browsers and web servers.
- HTTP is based on TCP, which means a TCP connection is established for each browser-server communication.

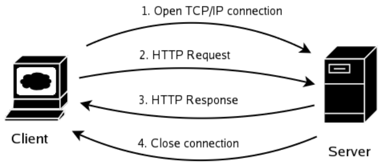
The Request-Response Cycle



A HTTP communication typically proceeds as follows.

1. The client opens a TCP connection to the server.

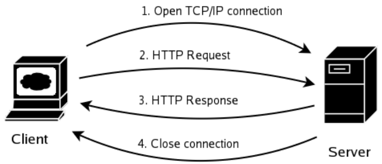
The Request-Response Cycle



A HTTP communication typically proceeds as follows.

1. The client opens a TCP connection to the server.
2. The client sends a request for a resource on the server. The request consists of a HTTP header, and data if the user submitted data to the server.

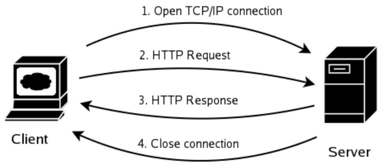
The Request-Response Cycle



A HTTP communication typically proceeds as follows.

1. The client opens a TCP connection to the server.
2. The client sends a request for a resource on the server. The request consists of a HTTP header, and data if the user submitted data to the server.
3. The server sends a response to the client. Also the response consists of HTTP headers, and data if the response required data.

The Request-Response Cycle



A HTTP communication typically proceeds as follows.

1. The client opens a TCP connection to the server.
2. The client sends a request for a resource on the server. The request consists of a HTTP header, and data if the user submitted data to the server.
3. The server sends a response to the client. Also the response consists of HTTP headers, and data if the response required data.
4. The server closes the TCP connection.

The Request-Response Cycle (Cont'd)

- HTTP is stateless. Neither server nor browser remembers anything about previous request-response cycles. Session handling must be added in server-side code.

The Request-Response Cycle (Cont'd)

- HTTP is stateless. Neither server nor browser remembers anything about previous request-response cycles. Session handling must be added in server-side code.
- To establish a TCP connection is expensive. Therefore, TCP connections might be kept alive and reused for multiple request-response cycles. This is specified with the **`keep-alive`** HTTP header, se below.

Cookies

- A cookie is a piece of data that is stored on the client.

Cookies

- A cookie is a piece of data that is stored on the client.
- The cookie is tagged with the server's domain name and included in every request to that server.

Cookies

- A cookie is a piece of data that is stored on the client.
- The cookie is tagged with the server's domain name and included in every request to that server.
- This enables the server to associate data with a specific client.

Cookies

- A cookie is a piece of data that is stored on the client.
- The cookie is tagged with the server's domain name and included in every request to that server.
- This enables the server to associate data with a specific client.
- Cookies can be used to store the user's settings, for example display language.

HTTP Sessions

- As mentioned above, HTTP is stateless.

HTTP Sessions

- As mentioned above, HTTP is stateless.
- Still, the server must be able to recognize which calls originate from the same client. Otherwise for example log in is impossible.

HTTP Sessions

- As mentioned above, HTTP is stateless.
- Still, the server must be able to recognize which calls originate from the same client. Otherwise for example log in is impossible.
- One commonly used method to solve this problem is to use cookies.

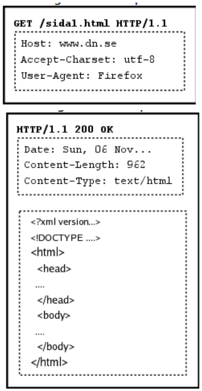
HTTP Sessions

- As mentioned above, HTTP is stateless.
- Still, the server must be able to recognize which calls originate from the same client. Otherwise for example log in is impossible.
- One commonly used method to solve this problem is to use cookies.
- If a request has a cookie with a session identifier, it identifies the user. If there is no such cookie, the user does not have a running session.

HTTP Sessions

- As mentioned above, HTTP is stateless.
- Still, the server must be able to recognize which calls originate from the same client. Otherwise for example log in is impossible.
- One commonly used method to solve this problem is to use cookies.
- If a request has a cookie with a session identifier, it identifies the user. If there is no such cookie, the user does not have a running session.
- On the server, the session id can be associated with any amount of data related to the user with that session. This is called conversational state.

HTTP Message Format

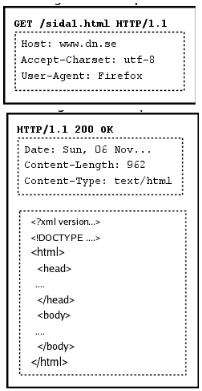


```
GET /sida1.html HTTP/1.1
Host: www.dn.se
Accept-Charset: utf-8
User-Agent: Firefox
```

```
HTTP/1.1 200 OK
Date: Sun, 06 Nov...
Content-Length: 962
Content-Type: text/html

<?xml version...>
<!DOCTYPE ....>
<html>
 <head>
 ....
 </head>
 <body>
 ....
 </body>
</html>
```

- A HTTP message has start-line, headers and body.

HTTP Message Format

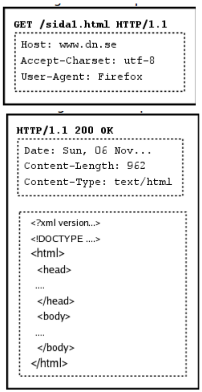


```
GET /sida1.html HTTP/1.1
Host: www.dn.se
Accept-Charset: utf-8
User-Agent: Firefox
```

```
HTTP/1.1 200 OK
Date: Sun, 06 Nov...
Content-Length: 962
Content-Type: text/html

<?xml version...>
<!DOCTYPE ....>
<html>
  <head>
  ....
  </head>
  <body>
  ....
  </body>
</html>
```

- A HTTP message has start-line, headers and body.
- The request start-line consists of HTTP method (se left), URL path and HTTP version, e.g., **`GET /page1.html HTTP/1.1`**

HTTP Message Format

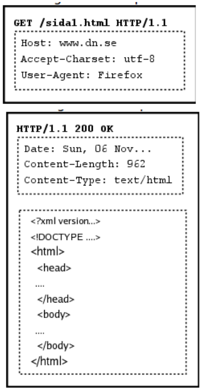


```
GET /sida1.html HTTP/1.1
Host: www.dn.se
Accept-Charset: utf-8
User-Agent: Firefox
```

```
HTTP/1.1 200 OK
Date: Sun, 06 Nov...
Content-Length: 962
Content-Type: text/html

<?xml version...>
<!DOCTYPE ....>
<html>
  <head>
  ....
  </head>
  <body>
  ....
  </body>
</html>
```

- A HTTP message has start-line, headers and body.
- The request start-line consists of HTTP method (se left), URL path and HTTP version, e.g., **`GET /page1.html HTTP/1.1`**
- The response start-line consists of HTTP version, status code and reason, e.g., **`HTTP/1.1 200 OK`**

HTTP Message Format



```
GET /sida1.html HTTP/1.1
Host: www.dn.se
Accept-Charset: utf-8
User-Agent: Firefox
```

```
HTTP/1.1 200 OK
Date: Sun, 06 Nov...
Content-Length: 962
Content-Type: text/html

<?xml version...>
<!DOCTYPE ....>
<html>
  <head>
  ....
  </head>
  <body>
  ....
  </body>
</html>
```

- A HTTP message has start-line, headers and body.
- The request start-line consists of HTTP method (se left), URL path and HTTP version, e.g., **`GET /page1.html HTTP/1.1`**
- The response start-line consists of HTTP version, status code and reason, e.g., **`HTTP/1.1 200 OK`**
- Sample request (top) and response (bottom) messages are depicted to the left.

Status Codes

- A HTTP response contains a status code to indicate the outcome of the request. There are five different categories of status codes.

  1xx Reply contains information, for example **`101`**, Switch Protocol.

Status Codes

- A HTTP response contains a status code to indicate the outcome of the request. There are five different categories of status codes.

1xx	Reply contains information, for example **`101`**, Switch Protocol.
2xx	Success, for example **`200`**, OK.

Status Codes

- A HTTP response contains a status code to indicate the outcome of the request. There are five different categories of status codes.

1xx	Reply contains information, for example **`101`**, Switch Protocol.
2xx	Success, for example **`200`**, OK.
3xx	Redirection, for example **`301`**, Moved Permanently.

Status Codes

- A HTTP response contains a status code to indicate the outcome of the request. There are five different categories of status codes.

1xx	Reply contains information, for example **`101`**, Switch Protocol.
2xx	Success, for example **`200`**, OK.
3xx	Redirection, for example **`301`**, Moved Permanently.
4xx	Client error, for example **`404`**, Not Found.

Status Codes

- A HTTP response contains a status code to indicate the outcome of the request. There are five different categories of status codes.

1xx	Reply contains information, for example **`101`**, Switch Protocol.
2xx	Success, for example **`200`**, OK.
3xx	Redirection, for example **`301`**, Moved Permanently.
4xx	Client error, for example **`404`**, Not Found.
5xx	Server error, for example **`500`**, Internal Server Error

HTTP Methods

- HTTP 1.1 has eight different methods that requires the following server actions.

GET	Deliver resource identified by the specified URL.

HTTP Methods

- HTTP 1.1 has eight different methods that requires the following server actions.

GET	Deliver resource identified by the specified URL.
POST	Accept message body and deliver it to the resource at the specified URL.

HTTP Methods

- HTTP 1.1 has eight different methods that requires the following server actions.

GET	Deliver resource identified by the specified URL.
POST	Accept message body and deliver it to the resource at the specified URL.
PUT	Accept message body and store it as a resource with the specified URL.

HTTP Methods

- HTTP 1.1 has eight different methods that requires the following server actions.

GET	Deliver resource identified by the specified URL.
POST	Accept message body and deliver it to the resource at the specified URL.
PUT	Accept message body and store it as a resource with the specified URL.
DELETE	Delete the resource at the given URL.

HTTP Methods

- HTTP 1.1 has eight different methods that requires the following server actions.

GET	Deliver resource identified by the specified URL.
POST	Accept message body and deliver it to the resource at the specified URL.
PUT	Accept message body and store it as a resource with the specified URL.
DELETE	Delete the resource at the given URL.
HEAD	Like GET, but only deliver headers.

HTTP Methods

- HTTP 1.1 has eight different methods that requires the following server actions.

GET	Deliver resource identified by the specified URL.
POST	Accept message body and deliver it to the resource at the specified URL.
PUT	Accept message body and store it as a resource with the specified URL.
DELETE	Delete the resource at the given URL.
HEAD	Like GET, but only deliver headers.
TRACE	Return the request message.

HTTP Methods

- HTTP 1.1 has eight different methods that requires the following server actions.

GET	Deliver resource identified by the specified URL.
POST	Accept message body and deliver it to the resource at the specified URL.
PUT	Accept message body and store it as a resource with the specified URL.
DELETE	Delete the resource at the given URL.
HEAD	Like GET, but only deliver headers.
TRACE	Return the request message.
OPTIONS	Tell which HTTP methods can be used with the specified URL.

HTTP Methods

- HTTP 1.1 has eight different methods that requires the following server actions.

GET	Deliver resource identified by the specified URL.
POST	Accept message body and deliver it to the resource at the specified URL.
PUT	Accept message body and store it as a resource with the specified URL.
DELETE	Delete the resource at the given URL.
HEAD	Like GET, but only deliver headers.
TRACE	Return the request message.
OPTIONS	Tell which HTTP methods can be used with the specified URL.
CONNECT	Connect to another host.

HTTP Methods

- HTTP 1.1 has eight different methods that requires the following server actions.

GET	Deliver resource identified by the specified URL.
POST	Accept message body and deliver it to the resource at the specified URL.
PUT	Accept message body and store it as a resource with the specified URL.
DELETE	Delete the resource at the given URL.
HEAD	Like GET, but only deliver headers.
TRACE	Return the request message.
OPTIONS	Tell which HTTP methods can be used with the specified URL.
CONNECT	Connect to another host.

- **GET** and **POST** are the most common methods and the only ones we will use in this course.

Safe and Idempotent Methods

- **`GET`** and **`HEAD`** are safe methods, which means they should not take any action other than to retrieve the specified resource.

Safe and Idempotent Methods

- **GET** and **HEAD** are safe methods, which means they should not take any action other than to retrieve the specified resource.
- **GET**, **HEAD**, **PUT**, **DELETE**, **OPTIONS** and **TRACE** are idempotent methods, which means the same request can be sent multiple times without any side-effects on the server.

Safe and Idempotent Methods

- **GET** and **HEAD** are safe methods, which means they should not take any action other than to retrieve the specified resource.
- **GET**, **HEAD**, **PUT**, **DELETE**, **OPTIONS** and **TRACE** are idempotent methods, which means the same request can be sent multiple times without any side-effects on the server.
- **POST** is not idempotent. If you submit the same purchase order multiple times in a web shop you will probably by multiple items. The purchase is typically a **POST** request.

When to Use GET

- Use **`GET`** when
  - The only desired action is to retrieve the specified resource.

When to Use GET

- Use **GET** when
  - The only desired action is to retrieve the specified resource.
  - If it shall be possible to bookmark the link.

When to Use GET

- Use **GET** when
  - The only desired action is to retrieve the specified resource.
  - If it shall be possible to bookmark the link.
  - The URL is shorter than 255 bytes. Note that a **GET** URL is longer than a **POST** URL since data is included in the URL which **GET**, but is in the message body with **POST** (see below).

When to Use GET

- Use **GET** when
  - The only desired action is to retrieve the specified resource.
  - If it shall be possible to bookmark the link.
  - The URL is shorter than 255 bytes. Note that a **GET** URL is longer than a **POST** URL since data is included in the URL which **GET**, but is in the message body with **POST** (see below).
  - You want to be able to write the entire request, including data, in the browser. This is useful when debugging.

When to Use POST

- Use **`POST`** when
  - The required action updates server state, for example saves something in a database.

When to Use POST

- Use **`POST`** when
  - The required action updates server state, for example saves something in a database.
  - The data does not fit within the 255 byte limit for URLs.

When to Use POST

- Use **`POST`** when
  - The required action updates server state, for example saves something in a database.
  - The data does not fit within the 255 byte limit for URLs.
  - The data shall not appear in the URL. Note that this is not a matter of security, data is sent in clear text also when using **`POST`**.

HTTP Parameters

- HTTP parameters are data included in a request to a web server.

HTTP Parameters

- HTTP parameters are data included in a request to a web server.
- A typical example is when the user has entered data in a HTML form.

len

HTTP Parameters

- HTTP parameters are data included in a request to a web server.
- A typical example is when the user has entered data in a HTML form.
- When using the **`GET`** method, parameters are appended to the URL as a query string, **`http://some.domain/some/path?city=stockholm&country=sweden`**

HTTP Parameters

- HTTP parameters are data included in a request to a web server.
- A typical example is when the user has entered data in a HTML form.
- When using the **`GET`** method, parameters are appended to the URL as a query string, **`http://some.domain/some/path?city=stockholm&country=sweden`**
- When using the **`POST`** method, parameters are included in the message body.

HTTP Headers

- HTTP **headers** have the syntax
  `name: value`

HTTP Headers

- HTTP headers have the syntax
  `name: value`
- There are several predefined headers, and it is also allowed to add new headers.

HTTP Headers

- HTTP headers have the syntax
  `name: value`
- There are several predefined headers, and it is also allowed to add new headers.
- Sample request headers are:

HTTP Headers

- HTTP headers have the syntax
  `name: value`
- There are several predefined headers, and it is also allowed to add new headers.
- Sample request headers are:

Host	The receiver address or domain name.

HTTP Headers

- HTTP headers have the syntax
  `name: value`
- There are several predefined headers, and it is also allowed to add new headers.
- Sample request headers are:

Host	The receiver address or domain name.
User-Agent	Identifies the sender browser and operating system.

HTTP Headers

- HTTP headers have the syntax
  `name: value`
- There are several predefined headers, and it is also allowed to add new headers.
- Sample request headers are:

Host	The receiver address or domain name.
User-Agent	Identifies the sender browser and operating system.
Content-Length	Message body length in bytes.

HTTP Headers

- HTTP headers have the syntax
  `name: value`
- There are several predefined headers, and it is also allowed to add new headers.
- Sample request headers are:

Host	The receiver address or domain name.
User-Agent	Identifies the sender browser and operating system.
Content-Length	Message body length in bytes.
Connection	Keep connection open future requests.

HTTP Headers

- HTTP headers have the syntax
  `name: value`
- There are several predefined headers, and it is also allowed to add new headers.
- Sample request headers are:

Host	The receiver address or domain name.
User-Agent	Identifies the sender browser and operating system.
Content-Length	Message body length in bytes.
Connection	Keep connection open future requests.

- Sample response headers are:

HTTP Headers

- HTTP headers have the syntax
  `name: value`
- There are several predefined headers, and it is also allowed to add new headers.
- Sample request headers are:

Host	The receiver address or domain name.
User-Agent	Identifies the sender browser and operating system.
Content-Length	Message body length in bytes.
Connection	Keep connection open future requests.

- Sample response headers are:

Content-Length	Message body length in bytes.

HTTP Headers

- HTTP headers have the syntax
  `name: value`
- There are several predefined headers, and it is also allowed to add new headers.
- Sample request headers are:

Host	The receiver address or domain name.
User-Agent	Identifies the sender browser and operating system.
Content-Length	Message body length in bytes.
Connection	Keep connection open future requests.

- Sample response headers are:

Content-Length	Message body length in bytes.
Content-Type	Media Type (see below) of response.

Media Type

- Media Type (or MIME Type) defines message content. This tells the receiver how to interpret the data.

Media Type

- Media Type (or MIME Type) defines message content. This tells the receiver how to interpret the data.
- Some media types are:

text/html	HTML markup
text/plain	Plain text
image/png	A png image
video/ogg	A ogg video.

Web Browsers

- It is important to test the web application with all different browsers that shall be able to display it.

Web Browsers

- It is important to test the web application with all different browsers that shall be able to display it.
- Browsers behave differently, and you should expect that some break specifications.

Web Servers

- The web server can deliver static content and also call server-side programs, like PHP, Java or .NET programs.

Web Servers

- The web server can deliver static content and also call server-side programs, like PHP, Java or .NET programs.
- The most commonly used web server is apache, `https://httpd.apache.org/`

Web Servers

- The web server can deliver static content and also call server-side programs, like PHP, Java or .NET programs.
- The most commonly used web server is apache, `https://httpd.apache.org/`
- Other common web servers are nginx, **`http://wiki.nginx.org/Main`** and Microsoft IIS.

Web Servers (Cont'd)

- You need to install a web server on your laptop. All labs will be reported on your own laptop, there is no web server in ICT school where you can run all the labs.

Web Servers (Cont'd)

- You need to install a web server on your laptop. All labs will be reported on your own laptop, there is no web server in ICT school where you can run all the labs.
- It might take time to get the web server running. You are advised to start installing the web server now.

Section

Web Development Tools

- There are many tools that facilitates developing web applications.

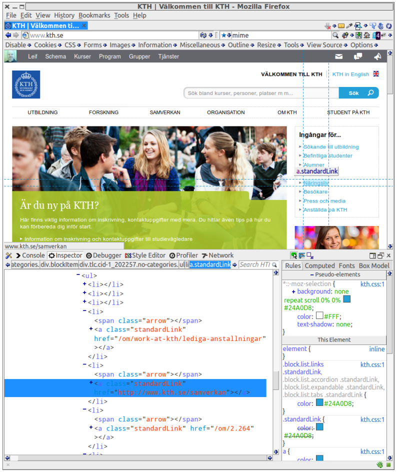
Web Development Tools

- There are many tools that facilitates developing web applications.
- Browser support varies between tools, most examples will be using Firefox.

Web Development Tools

- There are many tools that facilitates developing web applications.
- Browser support varies between tools, most examples will be using Firefox.
- You are strongly advised to start using some of the following tools, they will help you a lot.

Browser Web Console

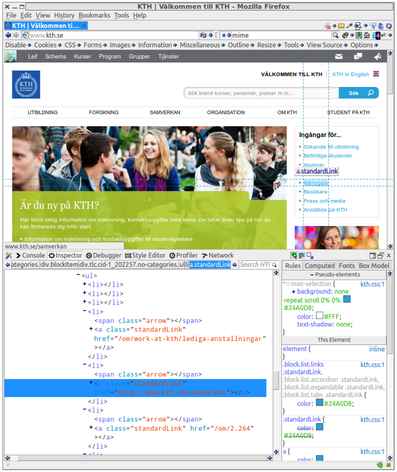

KTH | Välkommen till KTH - Mozilla Firefox
File Edit View History Bookmarks Tools Help
www.kth.se
Console Inspector Debugger Style Editor Profiler Network
ategories. div.blockItem div.tlc.cid-1_202257.no-categories. ul li a.standardLink
Rules Computed Fonts Box Model

```
- <ul>
  + <li></li>
  + <li></li>
  + <li></li>
  - <li>
      <span class="arrow"></span>
    + <a class="standardLink"
      href="/om/work-at-kth/lediga-anstallningar"
      ></a>
    </li>
  - <li>
      <span class="arrow"></span>
    + <a class="standardLink"
      href="http://www.kth.se/samverkan"></a>
    </li>
  - <li>
      <span class="arrow"></span>
    + <a class="standardLink" href="/om/2.264"
      ></a>
    </li>
```

```
*::-moz-selection {                         kth.css:1
    + background: none
repeat scroll 0% 0% #24A0D8;
    color: #FFF;
    text-shadow: none;
}
element {                                   inline
}
.block.list.links                           kth.css:1
.standardLink,
.block.list.accordion .standardLink,
.block.list.expandable .standardLink,
.block.list.tabs .standardLink {
    color: #24A0D8;
}
.standardLink {                             kth.css:1
    ~~color: #24A0D8;~~
}
a {                                         kth.css:1
```

- Most browsers have a built-in console.

Browser Web Console

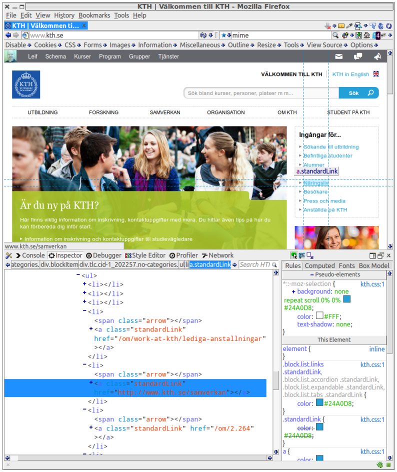

KTH | Välkommen till KTH - Mozilla Firefox
File Edit View History Bookmarks Tools Help
KTH | Välkommen ti...
www.kth.se
mime
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options
Leif Schema Kurser Program Grupper Tjänster
VÄLKOMMEN TILL KTH
KTH in English
Sök bland kurser, personer, platser m m...
Sök
UTBILDNING FORSKNING SAMVERKAN ORGANISATION OM KTH STUDENT PÅ KTH
Ingångar för...
Sökande till utbildning
Befintliga studenter
Alumner
a.standardLink
Näringsliv
Besökare
Press och media
Anställda på KTH
Är du ny på KTH?
Här finns viktig information om inskrivning, kontaktuppgifter med mera. Du hittar även tips på hur du kan förbereda dig inför start.
Information om inskrivning och kontaktuppgifter till studievägledare
www.kth.se/samverkan
Console Inspector Debugger Style Editor Profiler Network
ategories. div.blockItem div.tlc.cid-1_202257.no-categories. ul li a.standardLink
Search HTML
Rules Computed Fonts Box Model

```
- <ul>
  + <li></li>
  + <li></li>
  + <li></li>
  - <li>
      <span class="arrow"></span>
    + <a class="standardLink"
      href="/om/work-at-kth/lediga-anstallningar"
      ></a>
    </li>
  - <li>
      <span class="arrow"></span>
    + <a class="standardLink"
      href="http://www.kth.se/samverkan"></a>
    </li>
  - <li>
      <span class="arrow"></span>
    + <a class="standardLink" href="/om/2.264"
      ></a>
    </li>
```

```
– Pseudo-elements
*::-moz-selection {                         kth.css:1
    + background: none
repeat scroll 0% 0% 🟦
#24A0D8;
      color: □#FFF;
      text-shadow: none;
}
This Element
element {                                   inline
}
.block.list.links                           kth.css:1
.standardLink,
.block.list.accordion .standardLink,
.block.list.expandable .standardLink,
.block.list.tabs .standardLink {
      color: 🟦#24A0D8;
}
.standardLink {                             kth.css:1
      ~~color:~~ 🟦
~~#24A0D8;~~
}
a {                                         kth.css:1
      color: 🟦
```

- Most browsers have a built-in console.
- The console logs information associated with the web page, for example errors and warnings related to JavaScript, CSS and network requests.

Browser Web Console

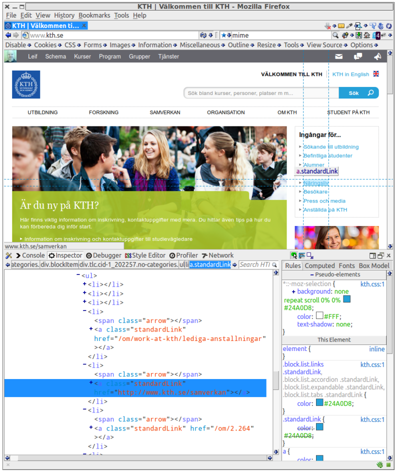

KTH | Välkommen till KTH - Mozilla Firefox
File Edit View History Bookmarks Tools Help
KTH | Välkommen ti...
www.kth.se
mime
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options
Leif Schema Kurser Program Grupper Tjänster
VÄLKOMMEN TILL KTH
KTH in English
Sök bland kurser, personer, platser m m...
Sök
UTBILDNING
FORSKNING
SAMVERKAN
ORGANISATION
OM KTH
STUDENT PÅ KTH
Ingångar för...
Sökande till utbildning
Befintliga studenter
Alumner
a.standardLink
Näringsliv
Besökare
Press och media
Anställda på KTH
Är du ny på KTH?
Här finns viktig information om inskrivning, kontaktuppgifter med mera. Du hittar även tips på hur du kan förbereda dig inför start.
Information om inskrivning och kontaktuppgifter till studievägledare
www.kth.se/samverkan
Console
Inspector
Debugger
Style Editor
Profiler
Network
ategories. div.blockItem div.tlc.cid-1_202257.no-categories. ul li a.standardLink
Search HTML
Rules
Computed
Fonts
Box Model
Pseudo-elements
This Element

```
- <ul>
  + <li></li>
  + <li></li>
  + <li></li>
  - <li>
      <span class="arrow"></span>
    + <a class="standardLink"
      href="/om/work-at-kth/lediga-anstallningar"
      ></a>
    </li>
  - <li>
      <span class="arrow"></span>
    + <a class="standardLink"
      href="http://www.kth.se/samverkan"></a>
    </li>
  - <li>
      <span class="arrow"></span>
    + <a class="standardLink" href="/om/2.264"
      ></a>
    </li>
```

```
*::-moz-selection {                      kth.css:1
    + background: none
repeat scroll 0% 0% 🟦
#24A0D8;
    color: ⬜#FFF;
    text-shadow: none;
}
element {                                   inline
}
.block.list.links                        kth.css:1
.standardLink,
.block.list.accordion .standardLink,
.block.list.expandable .standardLink,
.block.list.tabs .standardLink {
    color: 🟦#24A0D8;
}
.standardLink {                          kth.css:1
    ~~color: 🟦~~
~~#24A0D8;~~
}
a {                                      kth.css:1
    color: 🟦
```

- Most browsers have a built-in console.
- The console logs information associated with the web page, for example errors and warnings related to JavaScript, CSS and network requests.
- It enables you to run JavaScript expressions in the web page.

Browser Web Console


KTH | Välkommen till KTH - Mozilla Firefox
File Edit View History Bookmarks Tools Help
www.kth.se
Console Inspector Debugger Style Editor Profiler Network
ategories. div.blockItem div.tlc.cid-1_202257.no-categories. ul li a.standardLink
Rules Computed Fonts Box Model

```
- <ul>
  + <li></li>
  + <li></li>
  + <li></li>
  - <li>
      <span class="arrow"></span>
    + <a class="standardLink"
      href="/om/work-at-kth/lediga-anstallningar"
      ></a>
    </li>
  - <li>
      <span class="arrow"></span>
    + <a class="standardLink"
      href="http://www.kth.se/samverkan"></a>
    </li>
  - <li>
      <span class="arrow"></span>
    + <a class="standardLink" href="/om/2.264"
      ></a>
    </li>
```

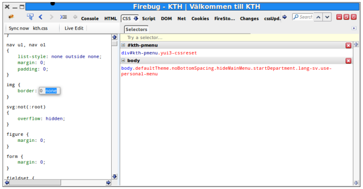
```
– Pseudo-elements
*::-moz-selection {                          kth.css:1
    + background: none
repeat scroll 0% 0% #24A0D8;
    color: #FFF;
    text-shadow: none;
}
This Element
element {                                    inline
}
.block.list.links                            kth.css:1
.standardLink,
.block.list.accordion .standardLink,
.block.list.expandable .standardLink,
.block.list.tabs .standardLink {
    color: #24A0D8;
}
.standardLink {                              kth.css:1
    ~~color: #24A0D8;~~
}
a {                                          kth.css:1
    color:
```

- Most browsers have a built-in console.
- The console logs information associated with the web page, for example errors and warnings related to JavaScript, CSS and network requests.
- It enables you to run JavaScript expressions in the web page.
- It also lets you choose elements from the web page and have their HTML and CSS displayed.

Browser (Cont'd)

- The console is opened with **`Ctrl-Shift-K`** in Firefox and **`Ctrl-Shift-J`** in Chrome.

Firebug

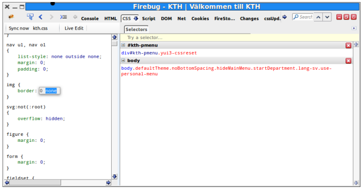

Firebug - KTH | Välkommen till KTH
Console HTML CSS Script DOM Net Cookies FireSto... Changes cssUpd.
Search
Sync now kth.css
Live Edit
Selectors
Try a selector...

```
nav ul, nav ol
{
    list-style: none outside none;
    margin: 0;
    padding: 0;
}

img {
    border: 0 none
}

svg:not(:root)
{
    overflow: hidden;
}

figure {
    margin: 0;
}

form {
    margin: 0;
}

fieldset {
```

```
#kth-pmenu
div#kth-pmenu.yui3-cssreset
body
body.defaultTheme.noBottomSpacing.hideMainMenu.startDepartment.lang-sv.use-
personal-menu
```

- Firebug is a powerful plug-in to Firefox.

Firebug

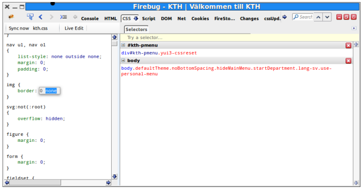


```
}

nav ul, nav ol
{
    list-style: none outside none;
    margin: 0;
    padding: 0;
}

img {
    border: 0 none
}

svg:not(:root)
{
    overflow: hidden;
}

figure {
    margin: 0;
}

form {
    margin: 0;
}

fieldset {
```

```
#kth-pmenu
div#kth-pmenu.yui3-cssreset
body
body.defaultTheme.noBottomSpacing.hideMainMenu.startDepartment.lang-sv.use-
personal-menu
```

- Firebug is a powerful plug-in to Firefox.
- In addition to console features, you can for example debug JavaScript, mark HTML elements, edit CSS and log network traffic.

Firebug

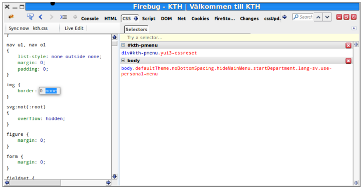


```
}

nav ul, nav ol
{
    list-style: none outside none;
    margin: 0;
    padding: 0;
}

img {
    border: 0 none
}

svg:not(:root)
{
    overflow: hidden;
}

figure {
    margin: 0;
}

form {
    margin: 0;
}

fieldset {
```

```
#kth-pmenu
div#kth-pmenu.yui3-cssreset
body
body.defaultTheme.noBottomSpacing.hideMainMenu.startDepartment.lang-sv.use-
personal-menu
```

- Firebug is a powerful plug-in to Firefox.
- In addition to console features, you can for example debug JavaScript, mark HTML elements, edit CSS and log network traffic.
- There are also many plug-ins to Firebug.

Firebug

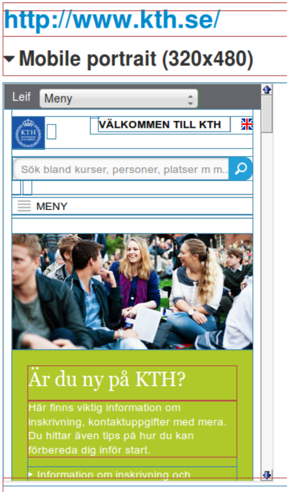

Firebug - KTH | Välkommen till KTH
Console HTML CSS Script DOM Net Cookies FireSto... Changes cssUpd.
Search
Sync now kth.css
Live Edit
Selectors
Try a selector...

```
nav ul, nav ol
{
    list-style: none outside none;
    margin: 0;
    padding: 0;
}

img {
    border: 0 none
}

svg:not(:root)
{
    overflow: hidden;
}

figure {
    margin: 0;
}

form {
    margin: 0;
}

fieldset {
```

```
#kth-pmenu
div#kth-pmenu.yui3-cssreset
body
body.defaultTheme.noBottomSpacing.hideMainMenu.startDepartment.lang-sv.use-
personal-menu
```

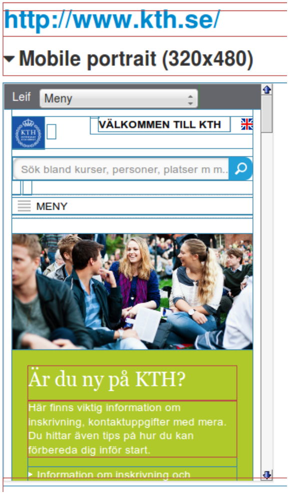
- Firebug is a powerful plug-in to Firefox.
- In addition to console features, you can for example debug JavaScript, mark HTML elements, edit CSS and log network traffic.
- There are also many plug-ins to Firebug.
- There is a cross-browser version of Firebug, written in JavaScript, that offers a subset of the functionality for most other browsers.

Web Developer


http://www.kth.se/
Mobile portrait (320x480)
Leif
Meny
VÄLKOMMEN TILL KTH
Sök bland kurser, personer, platser m m..
MENY
Är du ny på KTH?
Här finns viktig information om inskrivning, kontaktuppgifter med mera. Du hittar även tips på hur du kan förbereda dig inför start.
Information om inskrivning och

- Web Developer is a powerful plug-in to Firefox, which allows you to:

Web Developer


http://www.kth.se/

Mobile portrait (320x480)

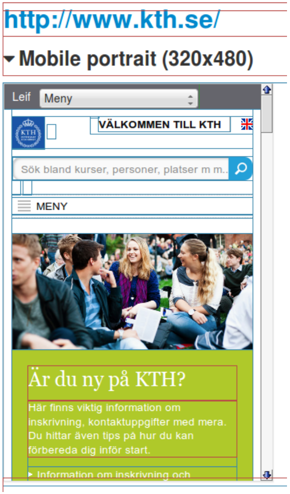
Är du ny på KTH?

Här finns viktig information om inskrivning, kontaktuppgifter med mera. Du hittar även tips på hur du kan förbereda dig inför start.

Information om inskrivning och

- Web Developer is a powerful plug-in to Firefox, which allows you to:
  - edit HTML and CSS.

Web Developer


http://www.kth.se/

Mobile portrait (320x480)

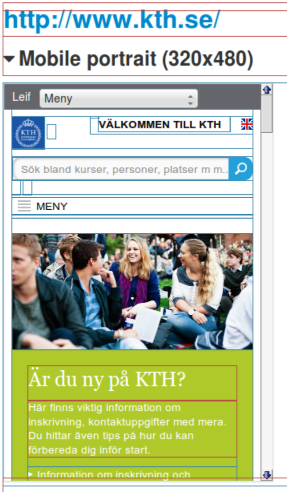
Är du ny på KTH?

Här finns viktig information om inskrivning, kontaktuppgifter med mera. Du hittar även tips på hur du kan förbereda dig inför start.

Information om inskrivning och

- Web Developer is a powerful plug-in to Firefox, which allows you to:
  - edit HTML and CSS.
  - See the area covered by a chosen element.

Web Developer

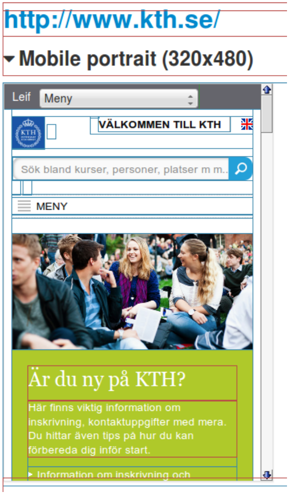

http://www.kth.se/
Mobile portrait (320x480)
Leif
Meny
VÄLKOMMEN TILL KTH
Sök bland kurser, personer, platser m m...
MENY
Är du ny på KTH?

Här finns viktig information om inskrivning, kontaktuppgifter med mera. Du hittar även tips på hur du kan förbereda dig inför start.

Information om inskrivning och

- Web Developer is a powerful plug-in to Firefox, which allows you to:
  - edit HTML and CSS.
  - See the area covered by a chosen element.
  - See the page in different screen resolutions.

Web Developer



http://www.kth.se/

Mobile portrait (320x480)

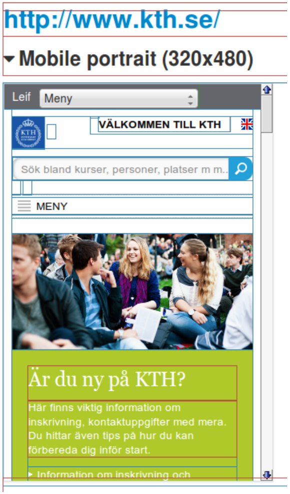
Är du ny på KTH?

Här finns viktig information om inskrivning, kontaktuppgifter med mera. Du hittar även tips på hur du kan förbereda dig inför start.

Information om inskrivning och

- Web Developer is a powerful plug-in to Firefox, which allows you to:
  - edit HTML and CSS.
  - See the area covered by a chosen element.
  - See the page in different screen resolutions.
  - Edit cookies.

Web Developer

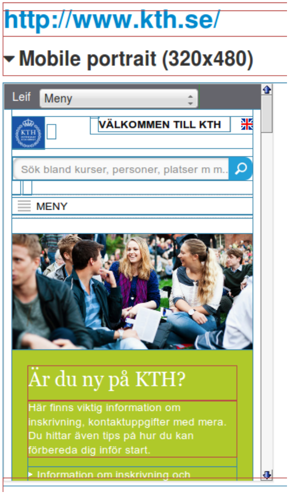

http://www.kth.se/
Mobile portrait (320x480)
Leif
Meny
VÄLKOMMEN TILL KTH
Sök bland kurser, personer, platser m m...
MENY
Är du ny på KTH?

Här finns viktig information om inskrivning, kontaktuppgifter med mera. Du hittar även tips på hur du kan förbereda dig inför start.

Information om inskrivning och

- Web Developer is a powerful plug-in to Firefox, which allows you to:
  - edit HTML and CSS.
  - See the area covered by a chosen element.
  - See the page in different screen resolutions.
  - Edit cookies.
  - Validate HTML and CSS.

Web Developer


http://www.kth.se/

Mobile portrait (320x480)

Är du ny på KTH?

Här finns viktig information om inskrivning, kontaktuppgifter med mera. Du hittar även tips på hur du kan förbereda dig inför start.

Information om inskrivning och

- Web Developer is a powerful plug-in to Firefox, which allows you to:
  - edit HTML and CSS.
  - See the area covered by a chosen element.
  - See the page in different screen resolutions.
  - Edit cookies.
  - Validate HTML and CSS.
- Web Developer has been ported to Chrome.

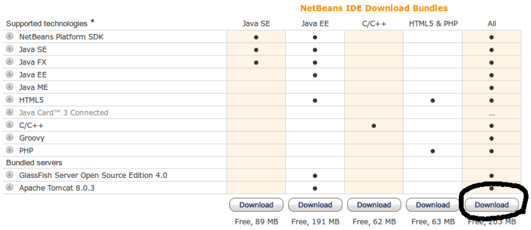
Validators

- There are online validators for both HTML and CSS. Links can be found on the course web site.

Validators

- There are online validators for both HTML and CSS. Links can be found on the course web site.
- Remember to always validate your HTML and CSS code.

NetBeans

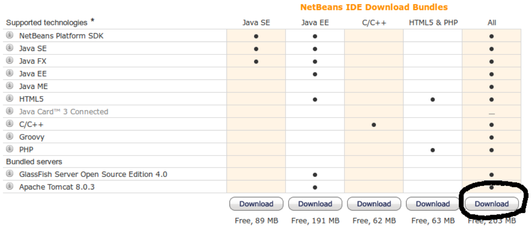


NetBeans IDE Download Bundles

Supported technologies *	Java SE	Java EE	C/C++	HTML5 & PHP	All
NetBeans Platform SDK	●	●			●
Java SE	●	●			●
Java FX	●	●			●
Java EE		●			●
Java ME					●
HTML5		●		●	●
Java Card™ 3 Connected					—
C/C++			●		●
Groovy					●
PHP				●	●
Bundled servers					
GlassFish Server Open Source Edition 4.0		●			●
Apache Tomcat 8.0.3		●			●
	Download	Download	Download	Download	Download
	Free, 89 MB	Free, 191 MB	Free, 62 MB	Free, 63 MB	Free, 203 MB

- There are many different IDEs for web development, all have their pros and cons.

NetBeans

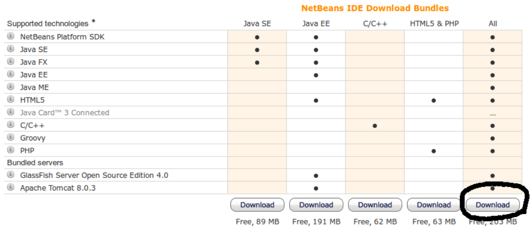


NetBeans IDE Download Bundles

Supported technologies *	Java SE	Java EE	C/C++	HTML5 & PHP	All
NetBeans Platform SDK	●	●			●
Java SE	●	●			●
Java FX	●	●			●
Java EE		●			●
Java ME					●
HTML5		●		●	●
Java Card™ 3 Connected					—
C/C++			●		●
Groovy					●
PHP				●	●
Bundled servers					
GlassFish Server Open Source Edition 4.0		●			●
Apache Tomcat 8.0.3		●			●
	Download	Download	Download	Download	Download
	Free, 89 MB	Free, 191 MB	Free, 62 MB	Free, 63 MB	Free, 203 MB

- There are many different IDEs for web development, all have their pros and cons.
- NetBeans will be used for examples during the course. Make sure to download the All version, see image above.

NetBeans



NetBeans IDE Download Bundles

Supported technologies *	Java SE	Java EE	C/C++	HTML5 & PHP	All
NetBeans Platform SDK	●	●			●
Java SE	●	●			●
Java FX	●	●			●
Java EE		●			●
Java ME					●
HTML5		●		●	●
Java Card™ 3 Connected					—
C/C++			●		●
Groovy					●
PHP				●	●
Bundled servers					
GlassFish Server Open Source Edition 4.0		●			●
Apache Tomcat 8.0.3		●			●
	Download	Download	Download	Download	Download
	Free, 89 MB	Free, 191 MB	Free, 62 MB	Free, 63 MB	Free, 203 MB

- There are many different IDEs for web development, all have their pros and cons.
- NetBeans will be used for examples during the course. Make sure to download the All version, see image above.
- Most important is that actually you use an IDE, do not program in a text editor unless you are really sure it is what you prefer.

JSFiddle and JSLint

- JSFiddle is an online editor where you can test HTML, CSS and JavaScript.

JSFiddle and JSLint

- JSFiddle is an online editor where you can test HTML, CSS and JavaScript.
- JSLint is an online tool for testing JavaScript code quality.

W3Schools Try It Yourself

- **`w3schools.com`** has excellent tutorials for all languages covered in the course.

W3Schools Try It Yourself

- `w3schools.com` has excellent tutorials for all languages covered in the course.
- All examples are presented with an online editor where you can experiment with your code.

Section

Use UI Guidelines!

- This is not a course in human-computer interaction. Still, it is mandatory to consider basic heuristics for user interface design.

Use UI Guidelines!

- This is not a course in human-computer interaction. Still, it is mandatory to consider basic heuristics for user interface design.
- There are some short introductory texts on user interface design available at Nielsen Norman Group, such as:

Use UI Guidelines!

- This is not a course in human-computer interaction. Still, it is mandatory to consider basic heuristics for user interface design.
- There are some short introductory texts on user interface design available at Nielsen Norman Group, such as:
  - 10 Usability Heuristics for User Interface Design, ***http://www.nngroup.com/articles/ten-usability-heuristics/***

Use UI Guidelines!

- This is not a course in human-computer interaction. Still, it is mandatory to consider basic heuristics for user interface design.
- There are some short introductory texts on user interface design available at Nielsen Norman Group, such as:
  - 10 Usability Heuristics for User Interface Design, `http://www.nngroup.com/articles/ten-usability-heuristics/`
  - Top 10 Mistakes in Web Design, `http://www.nngroup.com/articles/top-10-mistakes-web-design/`

Use UI Guidelines!

- This is not a course in human-computer interaction. Still, it is mandatory to consider basic heuristics for user interface design.
- There are some short introductory texts on user interface design available at Nielsen Norman Group, such as:
  - 10 Usability Heuristics for User Interface Design, `http://www.nngroup.com/articles/ten-usability-heuristics/`
  - Top 10 Mistakes in Web Design, `http://www.nngroup.com/articles/top-10-mistakes-web-design/`
  - Other lists linked from the latter.

Use UI Guidelines!

- This is not a course in human-computer interaction. Still, it is mandatory to consider basic heuristics for user interface design.
- There are some short introductory texts on user interface design available at Nielsen Norman Group, such as:
  - 10 Usability Heuristics for User Interface Design, *http://www.nngroup.com/articles/ten-usability-heuristics/*
  - Top 10 Mistakes in Web Design, *http://www.nngroup.com/articles/top-10-mistakes-web-design/*
  - Other lists linked from the latter.
- Here follows the 10 usability heuristics mentioned above.

J. Nielsen's UI Design Principles

1. The system should always keep users informed about what is going on.

J. Nielsen's UI Design Principles

1. The system should always keep users informed about what is going on.
2. Use words, phrases and concepts familiar to the user, rather than system-oriented terms.

J. Nielsen’s UI Design Principles

1. The system should always keep users informed about what is going on.
2. Use words, phrases and concepts familiar to the user, rather than system-oriented terms.
3. Implement undo and redo.

J. Nielsen’s UI Design Principles

1. The system should always keep users informed about what is going on.
2. Use words, phrases and concepts familiar to the user, rather than system-oriented terms.
3. Implement undo and redo.
4. Follow platform conventions, users should not have to wonder whether different words, situations, or actions mean the same thing.

J. Nielsen's UI Design Principles

5. Eliminate error-prone conditions or check for them and ask users to confirm before they commit to the action.

J. Nielsen’s UI Design Principles

5. Eliminate error-prone conditions or check for them and ask users to confirm before they commit to the action.
6. Minimize the user’s memory load by making objects, options, etc visible. The user should not have to remember information.

J. Nielsen's UI Design Principles

5. Eliminate error-prone conditions or check for them and ask users to confirm before they commit to the action.
6. Minimize the user's memory load by making objects, options, etc visible. The user should not have to remember information.
7. Use accelerators to speed up interaction for expert users.

J. Nielsen's UI Design Principles

5. Eliminate error-prone conditions or check for them and ask users to confirm before they commit to the action.
6. Minimize the user's memory load by making objects, options, etc visible. The user should not have to remember information.
7. Use accelerators to speed up interaction for expert users.
8. Remove irrelevant information.

J. Nielsen's UI Design Principles

9. Error messages should be expressed in plain language, precisely indicate the problem, and suggest a solution.

J. Nielsen’s UI Design Principles

9. Error messages should be expressed in plain language, precisely indicate the problem, and suggest a solution.
10. If necessary, provide help and documentation. The help should be easy to search, focused on the user’s task, and list concrete steps to be carried out.