Distributed Architectures

User Interface Design

Introduction to Internet Applications Internet Applications, ID1354

1/36



• Distributed Architectures

• User Interface Design

• Tools

Introduction

Distributed Architectures

Jser Interface Design

Distributed Architectures

Introduction

Jser Interface Design

Tools

• Distributed Architectures

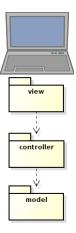
• User Interface Design

• Tools

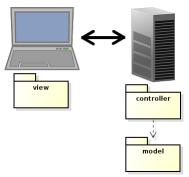
Section

Distributed Architectures

Local Application



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

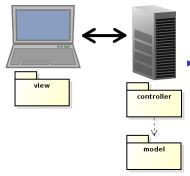


 Now, the application will be split on two tiers (computers).

Introduction

Distributed Architectures

Jser Interface Design

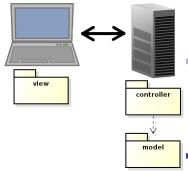


 Now, the application will be split on two tiers (computers).

 A client that has the view and a server that has controller and model. Introduction

Distributed Architectures

Jser Interface Design

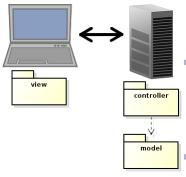


- Now, the application will be split on two tiers (computers).
 - A client that has the view and a server that has controller and model.
- The view is displayed in a web browser.

Introduction

Distributed Architectures

Jser Interface Design



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

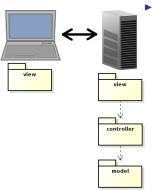
- A client that has the view and a server that has controller and model.
- The view is displayed in a web browser.

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

Introduction

Distributed Architectures

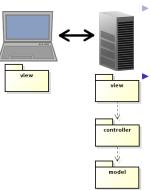
Jser Interface Design



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

Distributed Architectures

Jser Interface Design



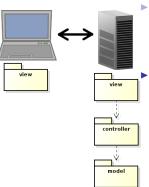
 First, we add a server layer, normally called view (a bit confusing).

However, the server side view layer performs tasks typical of a view:

Introduction

Distributed Architectures

Jser Interface Design



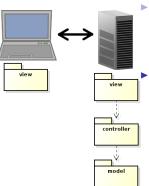
 First, we add a server layer, normally called view (a bit confusing).

However, the server side view layer performs tasks typical of a view:

 Creates views (HTML), which are sent to the client. Introduction

Distributed Architectures

Jser Interface Design



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

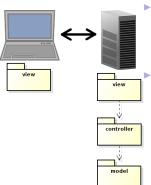
However, the server side view layer performs tasks typical of a view:

- Creates views (HTML), which are sent to the client.
- Interprets user gestures, a click in a web page creates a request to the server.

Introduction

Distributed Architectures

Jser Interface Design



 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.

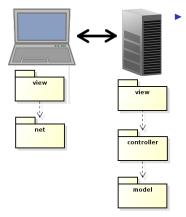
It might seem that we need yet a layer, for network handling. There is such a layer, but it is in the web server. We don't write it ourselves.

Introduction

Distributed Architectures

Jser Interface Design

Client-Side Communication



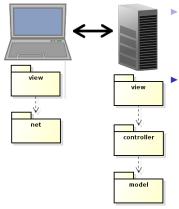
 Next, we add a client layer for communication, the net layer.

Introduction

Distributed Architectures

> Jser Interface Design

Client-Side Communication



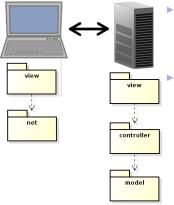
- Next, we add a client layer for communication, the net layer.
 - Actually, the browser handles most of the communication.
 - The small network code written by us is normally considered part of the client-side view, the net layer is omitted.

Introduction

Distributed Architectures

Jser Interface Design

Client-Side Communication

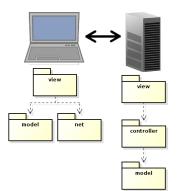


- Next, we add a client layer for communication, the net layer.
 - Actually, the browser handles most of the communication.
 - The small network code written by us is normally considered part of the client-side view, the net layer is omitted.
- This is a traditional web application.

Introduction

Distributed Architectures

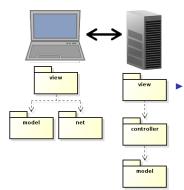
Jser Interface Design



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

Distributed Architectures

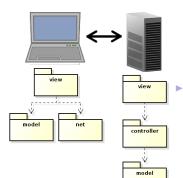
Jser Interface Design



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. Introduction

Jser Interface Design

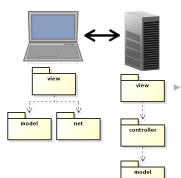


The trend is that data is stored also on the client, therefore we get a client-side model.

- 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.

Distributed Architectures

Jser Interface Design



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.

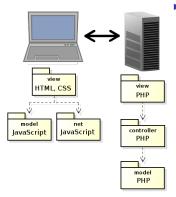
 Thereby, the application becomes faster.

 This is referred to as the MVVM, model-view-viewmodel pattern. Introduction

Distributed Architectures

Jser Interface Design

Programming Languages

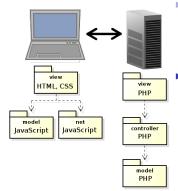


 This is the architecture we will normally use during the course. Introduction

Distributed Architectures

Jser Interface Design

Programming Languages



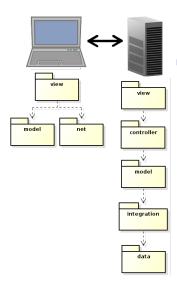
- This is the architecture we will normally use during the course.
- The view is programmed in HTML and CSS, client side behavior is programmed in JavaScript and the entire server side code is written in PHP.

Introduction

Distributed Architectures

Jser Interface Design

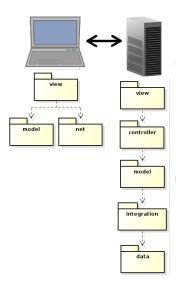
Three-Tier Architecture



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

> Jser Interface Design

Three-Tier Architecture



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

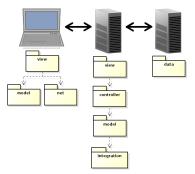
 We also introduce the integration layer, to handle the database calls.

Introduction

Distributed Architectures

Jser Interface Design

Three-Tier Architecture (Cont'd)



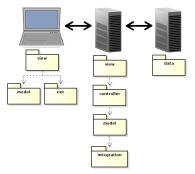
Distributed Architectures

> Jser Interface Design

Tools

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

Three-Tier Architecture (Cont'd)



Distributed Architectures

> Jser Interface Design

- 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.

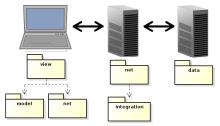
Distributed Architectures

Jser Interface Design

Tools

Question 1

Event-Driven Architecture



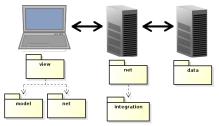
Distributed Architectures

> Jser Interface Design

Tools

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

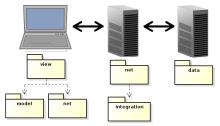




Jser Interface Design

- 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





Jser Interface Design

- 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.

Distributed Architectures

User Interface Design

Tools

Section

- Distributed Architectures
- User Interface Design
- Tools

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

Introduction

Distributed Architectures

User Interface Design

- 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:

Introduction

Distributed Architectures

User Interface Design

- 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 UI Design, http://www.nngroup.com/articles/ ten-usability-heuristics/

Introduction

Distributed Architectures

User Interface Design

- 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:
 - I0 Usability Heuristics for UI Design, http://www.nngroup.com/articles/ ten-usability-heuristics/
 - Top 10 Guidelines for Homepage Usability, http://www.nngroup.com/articles/ top-ten-guidelines-for-homepageusability/

Introduction

Distributed Architectures

User Interface Design

- 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:
 - I0 Usability Heuristics for UI Design, http://www.nngroup.com/articles/ ten-usability-heuristics/
 - Top 10 Guidelines for Homepage Usability, http://www.nngroup.com/articles/ top-ten-guidelines-for-homepageusability/
 - Top 10 Mistakes in Web Design, http://www.nngroup.com/articles/ top-10-mistakes-web-design/

Introduction

Distributed Architectures

User Interface Design

1. Visibility of system status

J. Nielsen's 10 Usability Heuristics

The system should always keep users informed about what is going on, through appropriate feedback. Introduction

Distributed Architectures

User Interface Design

1. Visibility of system status

J. Nielsen's 10 Usability Heuristics

- The system should always keep users informed about what is going on, through appropriate feedback.
- The UI must change within one second after a user action, or the user might think nothing happened.

Introduction

Distributed Architectures

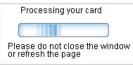
User Interface Design

1. Visibility of system status

J. Nielsen's 10 Usability Heuristics

- The system should always keep users informed about what is going on, through appropriate feedback.
- The UI must change within one second after a user action, or the user might think nothing happened.

Good examples:





Your password has been emailed.

Introduction

Distributed Architectures

User Interface Design

2. Match between system and the real world

- J. Nielsen's 10 Usability Heuristics
 - Use words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

Distributed Architectures

User Interface Design

2. Match between system and the real world

J. Nielsen's 10 Usability Heuristics

- Use words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
- Good example ("How can we help you?" better than "FAQ"):
 How can we help you?

Q Search help

Introduction

Distributed Architectures

User Interface Design

2. Match between system and the real world

J. Nielsen's 10 Usability Heuristics

- Use words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
- Good example ("How can we help you?" better than "FAQ"):
- Bad example ("Continue if enabled" is system oriented language):
 oops, there is a problem

Target.com requires cookie to be enabled.



Distributed Architectures

User Interface Design

3. User control and freedom

J. Nielsen's 10 Usability Heuristics

 We often do things by mistake, and therefore need a clearly marked "emergency exit" to leave an unwanted state without having to go through an extended dialogue. Support undo and redo. Introduction

Distributed Architectures

User Interface Design

3. User control and freedom

J. Nielsen's 10 Usability Heuristics

- We often do things by mistake, and therefore need a clearly marked "emergency exit" to leave an unwanted state without having to go through an extended dialogue. Support undo and redo.
- Good examples:

Sending... Cancel

Your message has been sent. Undo View message

Distributed Architectures

User Interface Design

4. Consistency and standards

J. Nielsen's 10 Usability Heuristics

 Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions. Introduction

Distributed Architectures

User Interface Design

4. Consistency and standards

J. Nielsen's 10 Usability Heuristics

- Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
- Good example (Sign in at top right, logo with link to index page at top left):



Distributed

Introduction

User Interface Design

5. Error prevention

J. Nielsen's 10 Usability Heuristics

Create a careful design which prevents problems from occurring. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action. Distributed Architectures

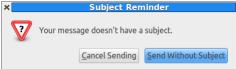
User Interface Design

5. Error prevention

J. Nielsen's 10 Usability Heuristics

Create a careful design which prevents problems from occurring. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

Good example:



Introduction

Distributed Architectures

User Interface Design

6. Recognition rather than recall

J. Nielsen's 10 Usability Heuristics

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. Introduction

Distributed Architectures

User Interface Design

6. Recognition rather than recall

J. Nielsen's 10 Usability Heuristics

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Good example:

Hotel search			
Destination, hotel, landmark o	r address		
Check in	Check out	_	_
13/08/2015 🔅	14/08/2015	8 1	
Thursday	Friday	Nig	ght
Rooms 1 room, 2 adults			
		Searc	h

- Clear headline.
- No doubt where to click to start the search.

Introduction

Distributed Architectures

User Interface Design

7. Flexibility and efficiency of use

J. Nielsen's 10 Usability Heuristics

Accelerators, unseen by the novice user, may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. Introduction

Distributed Architectures

User Interface Design

7. Flexibility and efficiency of use

J. Nielsen's 10 Usability Heuristics

Accelerators, unseen by the novice user, may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Examples are

- Saved searches
- Items you recently looked at
- Save query for later

Introduction

Distributed Architectures

User Interface Design

8. Aesthetic and minimalist design

J. Nielsen's 10 Usability Heuristics

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their visibility.

Introduction

Distributed Architectures

User Interface Design

8. Aesthetic and minimalist design

J. Nielsen's 10 Usability Heuristics

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their visibility.

Bad example:



Introduction

Distributed Architectures

User Interface Design

9. Help users recognize, diagnose, and recover from errors

J. Nielsen's 10 Usability Heuristics

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution. Introduction

Distributed Architectures

User Interface Design

9. Help users recognize, diagnose, and recover from errors

J. Nielsen's 10 Usability Heuristics

- Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
- Do not tell the user unexpected exception or anything similar.

Introduction

Distributed Architectures

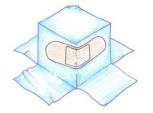
User Interface Design

9. Help users recognize, diagnose, and recover from errors

J. Nielsen's 10 Usability Heuristics

- Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
- Do not tell the user unexpected exception or anything similar.
- Good examples:

Email: *				
Telephone:				
D.O.B:	DD	MM	•	YYYY -



Error

Something went wrong. Don't worry, your files are still safe and the Dropboxers have been notified. Check out our Help Center and forums for help, or head back to home.

Introduction

Distributed Architectures

User Interface Design

10. Help and documentation

J. Nielsen's 10 Usability Heuristics

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large. Introduction

Distributed Architectures

User Interface Design

10. Help and documentation

- J. Nielsen's 10 Usability Heuristics
 - Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.
 - Good example:



Introduction

Distributed Architectures

User Interface Design

Distributed Architectures

User Interface Design

Tools

Question 2

Section

- Distributed Architectures
- User Interface Design
- Tools



Distributed Architectures

Jser Interface Design

Web Development Tools

 There are many tools that facilitates developing web applications. Introduction

Distributed Architectures

User Interface Design

Web Development Tools

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

Distributed Architectures

User Interface Design

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.

Distributed Architectures

Jser Interface Design



 Most browsers have a built-in console.

Introduction

Distributed Architectures

Jser Interface Design



- 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.

Introduction

Distributed Architectures

Jser Interface Design



- 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.

Introduction

Distributed Architectures

Jser Interface Design



- 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.

Introduction

Distributed Architectures

Jser Interface Design

Browser Web Console (Cont'd)

Distributed Architectures

Jser Interface Design

Tools

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

Firebug

🖉 📢 🗧 🗧 🔍 Cansole If	тня.	CIS & Script DOM Net Cesties FireSto Changes cutted. @ Please + v	800
ijncnow kth.cs Live Edit		Selectors	2
		Ty a selector	
<pre>we ul, nov el i iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii</pre>	3 Akth-pressu	8	
	61vRkth-pmena.yvd3-casreset		
macrico: D:		a boly	
pedding: 0;		body, defoultTheme.noBottonSpacing.hideHainHenu.startDepartment.lang-sw.use- personal.menu	
ng f border: 0.2002			
vg:not(:root)			
overflow: hidden;			
igure (margin: 0;			
mergin: 0;			
ielduer (a		

Distributed Architectures

Jser Interface Design

Tools

Firebug is a powerful plug-in to Firefox.

Firebug

Senc now kth.cs Live Edit		CIT & Sories DOM Net Cashies HireSto Changes catigat @ Pierri A V			
		Ty a selector.			
er ul. ave el	- Č	i dith-priori			
list-style: nese cutside none; margin: D:	civilith-prene vul3-cooreset				
	abotr				
pedding: 0;		body, defaultTheme.noDottonSpacing.hideMainNenu.startDepartment.lang-ss.use- personal-menu	-		
ng (border: 0.2002					
vg:not(:root)					
overflow: hidden;					
igure (margin: 0;					
ers { mergin: 0;					

Distributed Architectures

Jser Interface Design

- ► 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

P ♀ ♀ < > XE ♀ Cansole HT Services kth.cs	electers	Caules Press. Charp	s toops - prove the			
print Antas Che Los	Ty a selector					
er ul, nev el	Ty a selector					
er ut, nov et						
list-style: nose catside none;	Codkth-press.yc3-correct					
margin: 0;						
pediting: 0;	ty.defaultTheme.no8e rsemal-memu	tonSpacing.hide#ain#enu.s	tartDepartment.lang-sv.use-			
t border: 0						
wg:mot(:root)						
overflow: hidden;						
igare (margin: 0;						
wergin: 0;						

Distributed Architectures

Jser Interface Design

- ► 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

encnow Athicas Live Edit	B. CIS → Script DOM Net Coakies FireSto Changes coalignt ♥ Prover ▲ ♥	
	Selectors	
	Y a selector	_
ner ul, nov el	2 #kth-peneru	
list-style: neee outside none:	dowlikth-prenu, yud3-cooreset	
margin: 0;	a body	
pedding: 0;	body.defaultTheme.moBottonSpacing.hideMainWenu.startDepartment.lang-su.use- personal-menu	
torder: 0		
g:not(:root)		
overflow: hidden;		
igare (margin: 0;		
negin: 0;		

Distributed Architectures

Jser Interface Design

- ► 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.

http://www.kth.se/

Mobile portrait (320x480)

Leif Meny ‡	•
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	a

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

Introduction

Distributed Architectures

Jser Interface Design

http://www.kth.se/

Mobile portrait (320x480)

Leif Meny ‡	•
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	a

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

Introduction

Distributed Architectures

Jser Interface Design

http://www.kth.se/

Mobile portrait (320x480)

Leif Meny	•
	_
Sök bland kurser, personer, platser m m 👂	
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	4

- 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.

Introduction

Distributed Architectures

Jser Interface Design

http://www.kth.se/

Mobile portrait (320x480)

Leif Mer	ny	÷	4
١	VÄLKOMME	EN TILL KTH	*
Sök blan	nd kurser, personer,	, platser m m 🕽	0
	Y		_
Är d	lu ny på KT	Ή?	
inskrivr Du hitta	ns viktig informatior ning, kontaktuppgift ar även tips på hur eda dig inför start.		
 Inforr 	mation om inskrivnir	ng och	4

- 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.

Introduction

Distributed Architectures

Jser Interface Design

http://www.kth.se/

Mobile portrait (320x480)

Leif Meny		÷	•
® -	VÄLKOMME	N TILL KTH	*
Sök bland ku	irser, personer,	platser m m	<u>م</u>
MENY			_
Är du i	ny på KTI	H?	
	ktig information kontaktuppgifte		
Du hittar äv	en tips på hur d lig inför start.		
 Information 	on om inskrivnin	ŋ och	4

- 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.

Introduction

Distributed Architectures

Jser Interface Design

http://www.kth.se/

Mobile portrait (320x480)

Leif Meny ‡	<u>0</u>
	*
Sök bland kurser, personer, platser m m	୧
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 	ð

- 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.

Introduction

Distributed Architectures

Jser Interface Design

http://www.kth.se/

Mobile portrait (320x480)

Leif Meny	•
Sök bland kurser, personer, platser m m 🔎	
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.	
	æ

- 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.

Introduction

Distributed Architectures

Jser Interface Design



Distributed Architectures

User Interface Design

Tools

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

Distributed Architectures

Jser Interface Design

- 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.



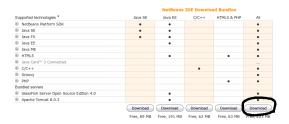
Distributed Architectures

User Interface Design

Tools

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

NetBeans



Distributed Architectures

> Jser Interface Design

- 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.

Distributed Architectures

User Interface Design

Tools

- 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 you actually use an IDE, do not program in a text editor unless you are really sure it is what you prefer.

NetBeans

JSFiddle and JSLint

Introduction

Distributed Architectures

Jser Interface Design

Tools

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

JSFiddle and JSLint

Introduction

Distributed Architectures

Jser Interface Design

- 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

Introduction

Distributed Architectures

Jser Interface Design

Tools

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.

Distributed Architectures

Jser Interface Design