



# Front-End I: HTML & CSS

## Language Technology and Web Applications

Johannes Graën, Nikolina Rajović, Igor Mustač<sup>1</sup>

Department of Computational Linguistics &  
Linguistic Research Infrastructure (LiRI)

September 27, 2023

---

<sup>1</sup>Parts of this lecture are based on the MDN web docs, which are licensed under CC-BY-SA 2.5 and on the book 'Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability' by Steve Krug.

## Learning Goals for this Week

- You recall the basic syntax of HTML and CSS
- You can write an HTML document with help of documentation
- You can write a CSS stylesheet with help of documentation
- You consider UX and accessibility aspects when developing

1. [HTML — Structuring a Website](#)
2. [CSS — Styling a Website](#)
3. [User experience and accessibility](#)

1. [HTML — Structuring a Website](#)
2. [CSS — Styling a Website](#)
3. [User experience and accessibility](#)

# What is HTML?

- A markup language which helps you to create and design web content
- XML mainly focuses on the transfer of data while HTML is focused on presentation
- HTML: Linking between documents (hyperlink)

```
<?xml version="1.0">
<address>
  <name>Krishna Rungta</name>
  <contact>9898613050</contact>
  <email>username@gmail.com</email>
  <birthdate>1985-09-27</birthdate>
</address>
```

```
<!DOCTYPE html>
<html>
<head>
  <title>Page title</title>
</head>
<body>
  <h1>First Heading</h1>
  <p>First paragraph.</p>
</body>
</html>
```

# A HTML Document

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title>My test page</title>
```

```
  </head>
```

```
  <body>
```

```
    <p>This is my page</p>
```

```
  </body>
```

```
</html>
```

```
<h1>This is a Title</h1>
```

```
<p>This is a paragraph.</p>
```

```
<h1>This is a Title</h1>
```

```
<p>This is a paragraph.</p>
```

---

# This is a Title

This is a paragraph.



## Anatomy of an HTML element



## Nesting elements

```
<p>My cat is <b>very</b> grumpy.</p>
```

```
<p>My cat is <strong>very</strong> grumpy.</p>
```

---

My cat is **very** grumpy.

My cat is **very** grumpy.

Attribute



The diagram shows the word "Attribute" with a vertical line pointing down to a horizontal bracket. The bracket is positioned above the `class="editor-note"` part of the HTML code below.

```
<p class="editor-note">My cat is very grumpy</p>
```

# Examples of Attributes

```
<h1 id="myheading">Title</h1>
```

```
<h1 class="main-heading green-heading">Title</h1>
```

# Examples of Attributes

```
<h1 id="myheading">Title</h1>
```

```
<h1 class="main-heading green-heading">Title</h1>
```

```

```

# Empty elements

`<p>`

Shall I compare thee to a summer's day?`<br>`

Thou art more lovely and more temperate:

`</p>`

---

Shall I compare thee to a summer's day?  
Thou art more lovely and more temperate:

# Whitespace in HTML

```
<p>Dogs are silly.</p>
```

```
<p>Dogs          are  
    silly.</p>
```

---

Dogs are silly.

Dogs are silly.

# Special Characters

- `&shy;`
- `< → &lt;`
- `> → &gt;`
- `" → &quot;`
- `' → &apos;`
- `& → &amp;`



# Special Characters

- `&shy;`
- `<` → `&lt;`
- `>` → `&gt;`
- `"` → `&quot;`
- `'` → `&apos;`
- `&` → `&amp;`

---

Example:

`<p>Use &lt;b>example text&lt;/b> to bold text</p>`

# Special Characters

- `&shy;`
- `<` → `&lt;`
- `>` → `&gt;`
- `"` → `&quot;`
- `'` → `&apos;`
- `&` → `&amp;`

---

Example:

`<p>Use &lt;b>example text</b> to bold text</p>`

---

Use `<b>example text</b>` to bold text

# Special Characters

- `&shy;`
- `<` → `&lt;`
- `>` → `&gt;`
- `"` → `&quot;`
- `'` → `&apos;`
- `&` → `&amp;`

---

Example:

`<p>Use &lt;b>example text&lt;/b> to bold text</p>`

---

Use `<b>`example text**`</b>`** to bold text

example1

```
<p>I'm not inside a comment</p>
```

```
<!-- I am! -->
```

## Indicating the primary language

```
<!-- On the document -->
```

```
<html lang="en-US">
```

# Indicating the primary language

*<!-- On the document -->*

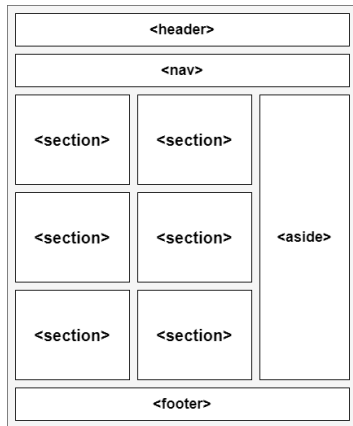
```
<html lang="en-US">
```

*<!-- On a single element -->*

```
<p lang="gsw">Grüezi!</p>
```

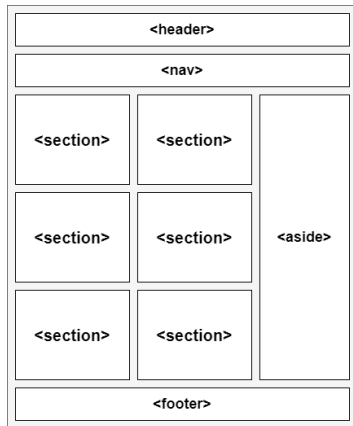
# HTML elements for structuring content

- Header: `<header>`
- Navigation bar: `<nav>`
- Main content: `<main>`
- Sections of main content: `<section>`
- Sidebar: `<aside>`; often placed inside `<main>`
- Footer: `<footer>`



# HTML elements for structuring content

- Header: `<header>`
- Navigation bar: `<nav>`
- Main content: `<main>`
- Sections of main content: `<section>`
- Sidebar: `<aside>`; often placed inside `<main>`
- Footer: `<footer>`





`<h1>Book Title</h1>`

`<p>By Jane Doe</p>`

`<h2>Chapter 1</h2>`

`<h3>A dark and stormy night</h3>`

---

# Book Title

By Jane Doe

## Chapter 1

A dark and stormy night

# Images

```
<!-- Different forms of source paths -->
```

```

```

```

```

```

```

```

```

```
<!-- Other image attributes -->
```

```

```

```
<ul>  
  <li>milk</li>  
  <li>eggs</li>  
  <li>bread</li>  
</ul>
```

- 
- milk
  - eggs
  - bread

```
<ol>  
  <li>Drive to the end of the road</li>  
  <li>Turn right</li>  
  <li>Go straight for two blocks</li>  
</ol>
```

- 
1. Drive to the end of the road
  2. Turn right
  3. Go straight for two blocks

```
<p>I am <em>glad</em> you weren't <em>late</em>.</p>
```

---

I am *glad* you weren't *late*.

<p>This liquid is <strong>highly toxic</strong>.</p>

---

This liquid is **highly toxic**.

```
<p>I'm creating a link to  
  <a href="https://www.uzh.ch/">the UZH homepage</a>.  
</p>
```

---

I'm creating a link to the UZH homepage.

## Turning other elements into hyperlinks

```
<a href="https://www.uzh.ch/">  
    
</a>
```



```
<h2 id="address">Mailing address</h2>
```

...

```
<!-- From the same page -->
```

Want to write us? Use our `<a href="#address">Address</a>`.

```
<!-- From a different page -->
```

Want to write us? Use our `<a href="/contact#address">Address</a>`.

## Inline elements and Block elements

```
<em>first</em><em>second</em><em>third</em>
```

```
<p>fourth</p><p>fifth</p><p>sixth</p>
```

---

*firstsecondthird*

fourth

fifth

sixth

# Non-semantic elements

- Block-level: `<div>`
- Inline: `<span>`

# Non-semantic elements

- Block-level: `<div>`
- Inline: `<span>`

Example usage:

`<p>`

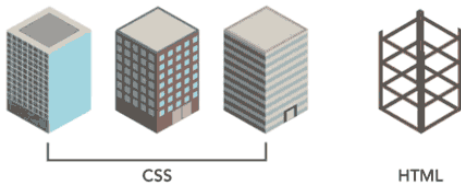
The King walked drunkenly back to his room at 01:00, the beer doing nothing to aid him as he staggered through the door

`<span class="editor-note">`[Editor's note: At this point in the play, the lights should be down low]`</span>`.

`</p>`

1. [HTML — Structuring a Website](#)
2. [CSS — Styling a Website](#)
3. [User experience and accessibility](#)

Cascading Style Sheets (CSS) is a stylesheet language that instructs the browser how we want our HTML to appear



```
h1 {  
    color: red;  
    font-size: 20px;  
}
```

```
p {  
    color: black;  
}
```

```
/* A comment */
```

# Linking an External Stylesheet

```
<html>
  <head>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    ...
  </body>
</html>
```



# Internal Stylesheets

```
<html>
  <head>
    ...

    <style>
      h1 {
        color: blue;
      }
      p {
        color: red;
      }
    </style>
  </head>
</html>
```

```
<p style="color:red; background-color:yellow">  
  Make this colorful!  
</p>
```

---

Make this colorful!

# Selectors: Tags

```
h1 {  
    /* ... */  
}
```

```
a {  
    /* ... */  
}
```

```
body {  
    /* ... */  
}
```

# Selectors: IDs and Classes

```
#myid {  
    /* ... */  
}
```

```
.myclass {  
    /* ... */  
}
```

```
p#myid {  
    /* ... */  
}
```

```
p.myclass {  
    /* ... */  
}
```

# Multiple Selectors

```
ul, ol, p.myclass {  
    color: green;  
}
```

## Setting a background

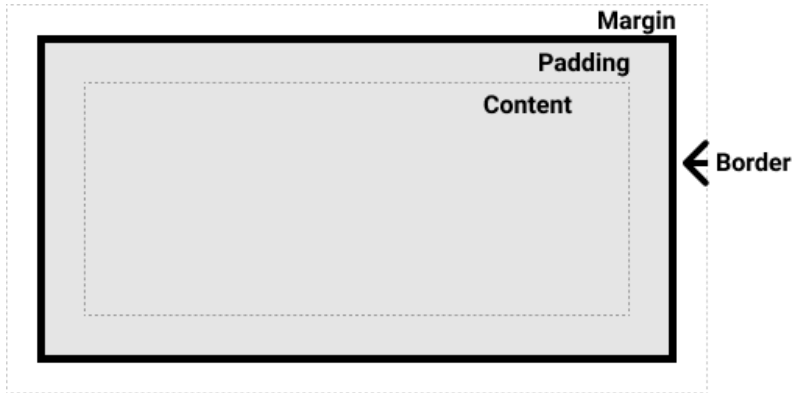
```
background: green;
```

```
background: rgb(255, 255, 128);
```

```
background: #74992e;
```

```
background: url("bg.jpg");
```

# CSS Box Model



# Spacing Examples

```
margin-left: 20px;
```

```
margin-left: 100%;
```

```
margin-left: 50vw;
```

```
margin-top: 50vh;
```



```
font-size: 12px;
```

```
font-size: 3em;
```

```
font-size: 2rem;
```

```
p {  
  font-family: Arial;  
}
```

*/\* Stacking fonts as a fallback \*/*

```
p {  
  font-family: "Helvetica Now", Helvetica, Arial, sans-serif;  
}
```

## Other useful text properties

```
font-style: italic;
```

```
font-weight: bold;
```

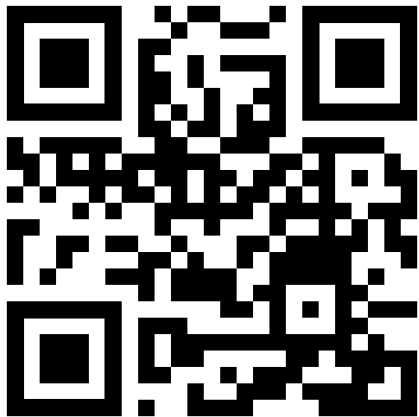
```
text-transform: uppercase;
```

```
text-decoration: underline;
```

```
text-align: left;  
text-align: right;  
text-align: center;  
text-align: justify;
```

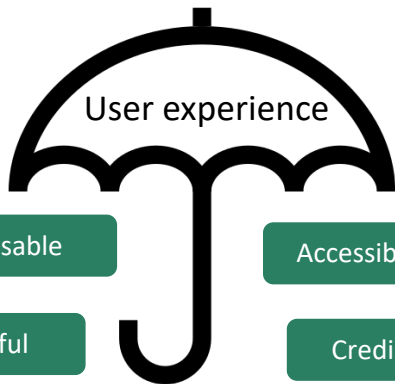
1. [HTML — Structuring a Website](#)
2. [CSS — Styling a Website](#)
3. [User experience and accessibility](#)

# What is user experience?



<https://t.ly/6RVo2>

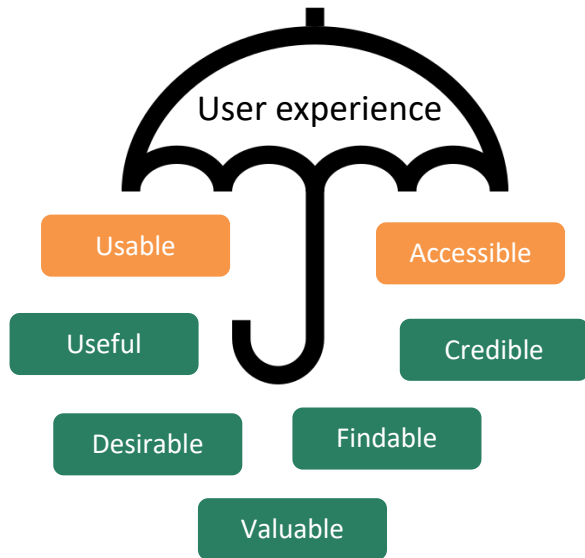
# What is user experience?



How people feel when using a service?

- Does it provide a value
  - Is it easy to use
  - Is it enjoyable
- 
- Designing with user's needs and expected behaviour in mind

# What is user experience?



How people feel when using a service?

- Does it provide a value
  - Is it easy to use
  - Is it enjoyable
- 
- Designing with user's needs and expected behaviour in mind



# Usability vs user experience

Usability by ISO 9421-11: "The extent to which a product can be used by specified users to achieve specified goals, with effectiveness, efficiency and satisfaction in a specified context of use."

5 criteria for usability by UXPA(User Experience Professionals Association):

Effectiveness

Error tolerance

Efficiency

Ease of learning

Engagement

## Usability testing -> Usability report

2015 joint research on B2B web users:

<http://komarketing.com/files/b2b-web-usability-report-2015.pdf>

- 46% left the page because they couldn't find what exactly the company does (effective message)
- 44% left because they couldn't find contact info (credibility)

## Usability testing -> Usability report

2015 joint research on B2B web users:

<http://komarketing.com/files/b2b-web-usability-report-2015.pdf>

- 46% left the page because they couldn't find what exactly the company does (effective message)
- 44% left because they couldn't find contact info (credibility)

How can anyone conduct usability test?

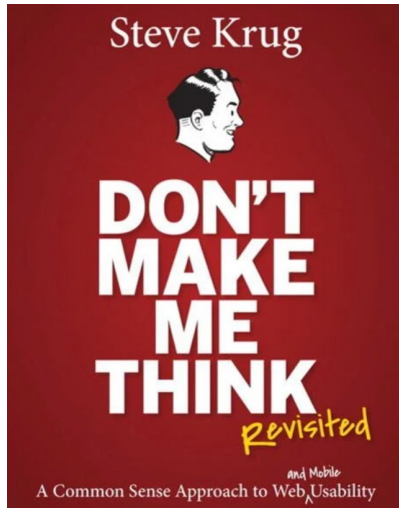
- 3-4 users per round
- Start early in the project
- Observe/record the screen
- "Don't throw the baby out with the dishes"

## Usability guidelines (1)

"It's not rocket surgery." TM

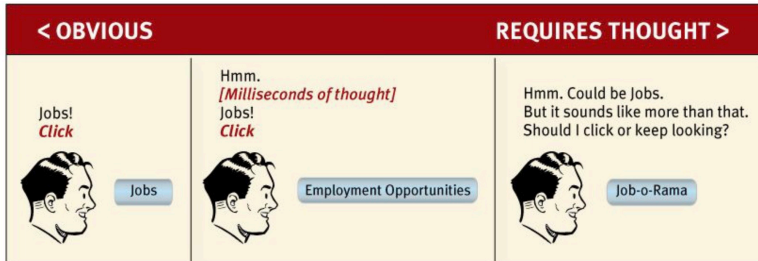
Just use Advanced Common Sense.

*Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability* - Steve Krug



# Usability guidelines (2)

## 1. "Don't make me think!"



Each page should be self-evident or at least self-explanatory  
(competition is usually one click away)

## Usability guidelines (3)

2. "Don't look now, but I think it's following us." - persistent navigation should be present on every page (except forms and on home page)
3. Logo or Site ID should be visible on every page
4. "There is no place like home."
5. A way to search the content – a box, a button and a word "Search"
6. "You are here." - highlighting of the current location (or breadcrumbs)



©2000. The New Yorker Collection from cartoonbank.-  
com. All Rights Reserved.

## Usability guidelines (4)

## 7. Create a clear visual hierarchy on each page

## 8. Take advantage of conventions

## 9. Break pages up into clearly defined areas

## "Banner blindness" - eye tracking study

## 10. Make it obvious what is clickable

## 11. Minimize noise and omit needless words

## 12. Conduct usability tests



# How we use websites?

## WHAT WE DESIGN FOR...

Read

Read

Read

Read

[Pause for reflection]

Finally, click on carefully chosen link

## THE REALITY...

Look around feverishly for anything that

a) is interesting, or vaguely resembles what you're looking for, and

b) is clickable.

As soon as you find a halfway-decent match, click.

If it doesn't pan out, click the Back button and try again.

- We don't read. We scan pages.
- We don't make optimal choices. We "satisfice".
- We don't figure out how things work. We muddle through
- Difference between expert users and average users



# Accessibility

- practice of making websites usable by as many people as possible
- people with visual impairments (screen magnifiers, screen readers)
- people with hearing impairments (text simplification, audio transcripts, subtitles)
- people with mobility impairments (keyboard accessibility, head pointers)
- people with cognitive impairments (plain-language standards, minimising distractions)
- we traditionally think of this as being about people with disabilities, but it also includes those using mobile devices, or those with slow network connections

[https://developer.mozilla.org/en-US/docs/Learn/Accessibility/What\\_is\\_accessibility](https://developer.mozilla.org/en-US/docs/Learn/Accessibility/What_is_accessibility)

*A Web for everyone: Designing Accessible User Experiences* - S. Horton, W.Quesenbery

# Accessibility guidelines

Things you can do right now to increase accessibility:

1. Fix the usability problems that confuse everyone
2. Read this old but relevant article about how blind people use screen readers: [https://redish.net/wp-content/uploads/Theorfanos\\_Redish\\_InteractionsPaperAuthorsVer.pdf](https://redish.net/wp-content/uploads/Theorfanos_Redish_InteractionsPaperAuthorsVer.pdf)
3. Implement specific changes in your pages like:
  - Add appropriate alt text to every image (and empty or "null" when image should be ignored by screen readers)
  - Use headings correctly (<h1> for main title, <h2> for major section headings, <h3> for subheadings)
  - Make your forms work with screen readers (use <label> element to associate fields with their labels)
  - Put a "Skip to Main Content" link at the beginning of every page
  - Make all content accessible by keyboard
  - Create significant contrast between your text and background



Questions?