

HTML & CSS

Summary

Boris.Fritscher@he-arc.ch

Web Development

- **HTML** for content
- **CSS** for styling
- **JavaScript** for logic

HTML

Hyper Text Markup Language

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Document Title</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1>Main heading in my document</h1>
    <p>Look Ma, I am coding <strong>HTML</strong>.</p>

    <!-- This is a comment -->
    <h2>Sub heading</h2>
    <p>Some more text</p>
  </body>
</html>
```

Main heading in my
document

Look Ma, I am coding **HTML**.

Sub heading

Some more text

HTML: Elements

Element	Description
<code><!DOCTYPE html></code>	Document type must be the very first thing in your HTML document
<code><html></code>	Root of an HTML document
<code><head></code>	Metadata section of the document
<code><title></code>	Defines the title of the document (only text)
<code><meta></code>	Provides machine parsable metadata about the HTML document
<code><body></code>	Contains the content of the document
<code><!-- --></code>	Comment everything between <code><</code> and <code>></code>
<code><h1></code> , <code><h2></code> , <code><h3></code> , <code><h4></code>	Heading most important to least
<code><p></code>	A paragraph of text
<code></code>	Give text strong importance, displayed in bold

HTML: Attributes

Elements can have attributes **name="value"**

Attribute	Description
< ... id="" ... >	All element can have an id, must be unique in the page!
< ... class="" ... >	All element can have a list of classes (used for CSS)
	Image from src with alt description
	Anchor, hyperlink to a target destination

```
<div id="page2">
  <p>A <span class="fun">funny</span> dog:
    <img src="" alt="" /></p>
  <p>go to <a href="page3.html">page 3</a></p>
</div>
```

Ids and classes are used by css and javascript to reference specific individual or groups of elements.

HTML: More Elements

Element	Description
<code><div></code>	A generic container for elements
<code><nav></code>	Represents a section of a page that links to other pages or parts
<code><section></code>	A thematic grouping of content
<code><header></code>	A page header container for logo, nav, search
<code><footer></code>	A page or section footer container
<code><pre></code>	Preformatted text, whitespace are displayed as typed
<code></code> , <code></code>	Unordered list and ordered list
<code></code>	List item inside a ul or ol
<code></code>	A generic inline container

HTML: Tables

```
<!DOCTYPE html>
<table>
  <caption>Caption of the table (optional)</caption>
  <thead>
    <tr>
      <th>Column A</th> <th>Column B</th> <th>Column C</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td> <td>2</td> <td>3</td>
    </tr>
    <tr>
      <td>4</td> <td colspan="2" rowspan="2">5</td>
    </tr>
    <tr>
      <td>6</td>
    </tr>
  </tbody>
</table>
```

Caption of the table (optional)

Column A	Column B	Column C
1	2	3
4	5	
6		

HTML: Forms

```
<form>
  <p><label for="n">Name: </label> <!-- two ways to associate labels -->
    <input id="n" name="name"
      type="text" placeholder="Enter your name"/></p>
  <p><label>Password: <input type="password" /></label></p>

  <p><input type="checkbox" name="on" value="1"/>on
  <!-- radio groups need same name -->
  <input type="radio" name="r" value="1" checked/>1
  <input type="radio" name="r" value="2"/>2</p>

  <p><textarea name="multiline"></textarea></p>

  <p><select name="choice">
    <option value="v1">Value 1</option>
    <option value="v2" selected>Value 2</option>
    <option value="v3">Value 3</option>
  </select></p>

  <p><button>Hello</button> <input type="submit" /></p>
</form>
```

Name:

Password:

☐ on ☒ 1 ☐ 2

Value 2 ▼

HTML: inline elements

An inline element occupies only the space bounded by the tags that define the inline element.

- may contain only data and other inline elements.
- do not begin with new line.

*strong, a, img, span, button, input, label,
select, textarea*

HTML: block-level elements

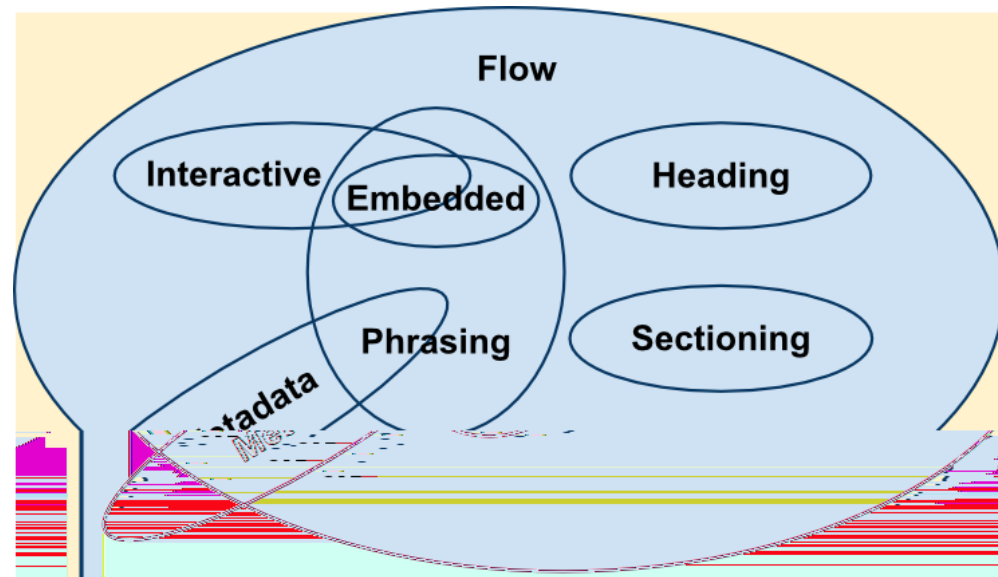
A block-level element occupies the entire space of its parent element (container), thereby creating a "block."

- block-level elements begin on new lines.

div, ul, form, h1, h2, header, nav, pre, p

HTML5: content categories

In HTML5, this binary distinction of **block-level** vs. **inline** elements is replaced with a more complex set of content categories



block-level \approx flow content, **inline** \approx phrasing content

HTML: including CSS and JavaScript

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>External files</title>
    <link href="css/style.css" rel="stylesheet">
  </head>
  <body>
    ...
    <script src="http://code.jquery.com/jquery.js"></script>
    <script src="js/myscript.js"></script>
  </body>
</html>
```

Later listings take precedence.

Script tags at the end in order to not block page loading.

Other script loading attributes

```
<script src="" defer></script>
```

script is meant to be executed after the document has been parsed, but before firing *DOMContentLoaded*

```
<script src="" async></script>
```

specifies that the script will be executed asynchronously as soon as it is available.

- If async is present: The script is executed asynchronously with the rest of the page (the script will be executed while the page continues the parsing)
- If async is not present and defer is present: The script is executed when the page has finished parsing
- If neither async or defer is present: The script is fetched and executed immediately, before the browser continues parsing the page

CSS

Cascading Style Sheets

CSS: Selectors

```
body {  
  width: 500px;  
}  
  
.tag-class {  
  padding: 20px 40px;  
  width: auto;  
}  
  
a:hover{  
  color: red;  
}  
  
#tag-id {  
  height: 100px;  
  float: left;  
}
```

element, class, pseudo-class, id

CSS Selector	Action
*	selects all elements
aa *	selects all elements within the aa elements.
aa bb	selects all bb elements within the aa elements.
aa, bb	selects all aa and bb elements, it's just a list.
aa.classname	selects all aa elements with the class classname.
#idname	selects the element with the id idname.
a:hover	selects the a element if the mouse hovers over it (pseudo-classes).
aa > bb	selects the bb element if it's a direct child of aa.
aa + bb	selects the bb element if it's immediately preceded by aa.
aa ~ bb	selects every bb element that are preceded by a aa element.
aa[attr="value"]	selects the aa element with a attr="value" attribute.
:first-child	selects all first child elements.
:nth-child(A)	selects every element that is the Ath child of another element.













Visualization: [CSS Visualizer](#)

Exercise: [CSS Diner](#)

CSS: Cascading Rules

http://www.stuffandnonsense.co.uk/archives/css_specificity_wars.html

AND ALL THAT
halarkey

 <p>a</p> <p>1 x element selector</p> <p>Sith: 0, 0, 1</p>	 <p>p a</p> <p>2 x element selectors</p> <p>Sith: 0, 0, 2</p>	 <p>.whatever</p> <p>1 x class selector</p> <p>Sith: 0, 1, 0</p>	 <p>a.whatever</p> <p>1 x element selector 1 x class selector</p> <p>Sith: 0, 1, 1</p>
 <p>p a.whatever</p> <p>2 x element selectors 1 x class selector</p> <p>Sith: 0, 1, 2</p>	 <p>.whatever .whatever</p> <p>2 x class selectors</p> <p>Sith: 0, 2, 0</p>	 <p>p.whatever a.whatever</p> <p>2 x element selectors 2 x class selectors</p> <p>Sith: 0, 2, 2</p>	 <p>#whatever</p> <p>1 x id selector</p> <p>Sith: 1, 0, 0</p>
 <p>a#whatever</p> <p>1 x element selector 1 x id selector</p> <p>Sith: 1, 0, 1</p>	 <p>.whatever a#whatever</p> <p>1 x element selectors 1 x class selector 1 x id selector</p> <p>Sith: 1, 1, 1</p>	 <p>.whatever .whatever #whatever</p> <p>2 x class selectors 1 x id selector</p> <p>Sith: 1, 2, 0</p>	 <p>#whatever #whatever</p> <p>2 x id selectors</p> <p>Sith: 2, 0, 0</p>

```
<!DOCTYPE html>
<html>
<body>
  <div id="b1" class="box">BOX1</div>
</body>
</html>
```

```
#b1.box{
  color: yellow;
}
```

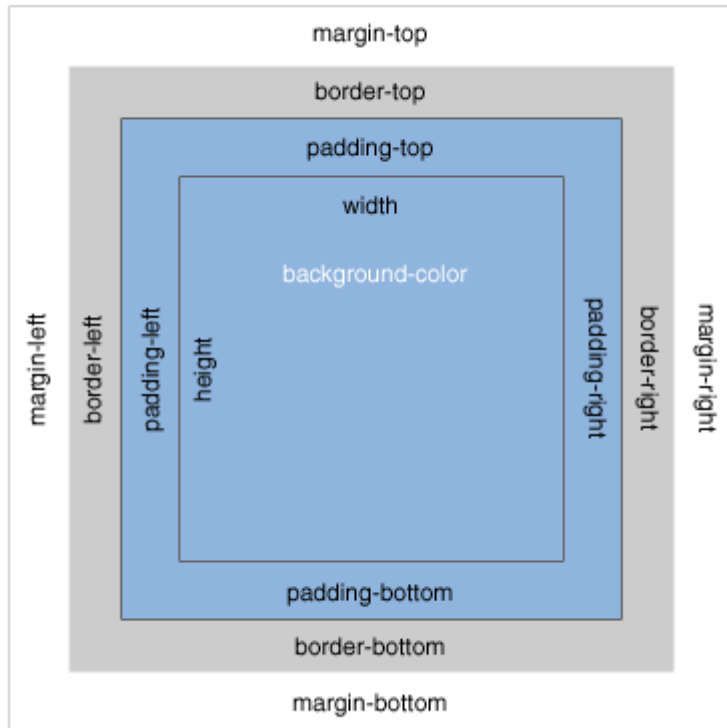
```
#b1.box{
  color: pink;
}
```

```
.box{
  color: green;
}
```

```
#b1 {
  color: blue;
}
```

```
div {
  color: red;
}
```

CSS: box-model



box-sizing: content-box; default

box-sizing: border-box; width includes padding + border

Collapsing margins: depending on setting a border or padding, margins are collapsed or put apart. Normally, bottom and top margin margins are merged together.

center a block element: *margin: 0 auto;*

CSS: border

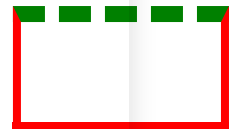
```
border-width: 4px;  
border-color: red;  
border-style: solid;
```

```
border-top-width: 8px;  
border-top-color: green;  
border-top-style: dashed;
```

```
/* shorthand */  
border: 2px solid #0000FF;
```

```
border-left: 10px solid #FFFF00;
```

```
border-radius: 10px;
```



- <https://developer.mozilla.org/en-US/docs/Web/CSS/border-radius?v=example>
- <http://css3pie.com/demos/border-radius/>
- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Background_and_Borders/Border-radius_generator

CSS background

```
background-color: [#color, gradient];  
background-image: url('../image.jpg');  
background-position: x y;  
background-repeat: [repeat, repeat-x, repeat-y, no-repeat, space, round];  
background-size: [auto, x y, cover, contain];  
  
background-clip: [border-box, padding-box, content-box, inherit];  
background-origin: [border-box, padding-box, content-box, inherit];  
background-attachment: [scroll, fixed, local];
```

- <https://css-tricks.com/almanac/properties/b/background/>
- <http://www.colorzilla.com/gradient-editor/>

CSS text

```
font-family: [Times New Roman, serif, Arial, sans-serif, Consolas, monospace];  
font-style: [normal, italic];  
font-weight: [normal, bold, 100-900];  
font-size: 20px;  
line-height: 30px;
```

Times New Roman, Arial, Consolas, Open Sans and many more with [WebFonts](#)

CSS text

```
font-family: [Times New Roman, serif, Arial, sans-serif, Consolas, monospace];  
font-style: [normal, italic];  
font-weight: [normal, bold, 100-900];  
font-size: 20px;  
line-height: 30px;
```

Times New Roman, Arial, Consolas, Open Sans and many more with [WebFonts](#)

... and crazy text-shadow

CSS text

```
font-family: [Times New Roman, serif, Arial, sans-serif, Consolas, monospace];  
font-style: [normal, italic];  
font-weight: [normal, bold, 100-900];  
font-size: 20px;  
line-height: 30px;
```

Times New Roman, Arial, Consolas, Open Sans and many more with [WebFonts](#)

... and crazy text-shadow

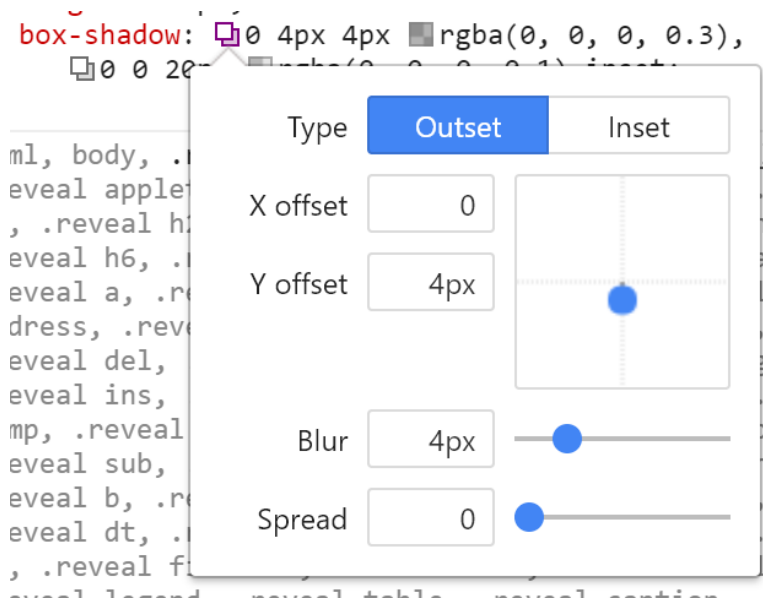
- <https://css-tricks.com/almanac/properties/t/text-shadow/>

CSS box-shadow



Use chrome dev-tools editor to adjust box-shadow.

- <https://css-tricks.com/almanac/properties/b/box-shadow/>



CSS: Positioning

`position: static|fixed|absolute|relative;`

- **static** : default not positioned in any special way
- **fixed** : positioned relative to the viewport
- **absolute** : behaves like *fixed* except relative to the nearest positioned ancestor
- **relative** : behaves the same as static unless you add some extra properties.

`top|right|bottom|left: <<units>>;`

`z-index: <<int>>;` *change ordering of positionned elements*

`float: left|right;`

`clear: left|right|both;` *disallow float*

```
<header>
  <nav>menu</nav>
</header>
<div id="s1" class="square"></div>
<div id="s2" class="square"></div>
<div id="s3" class="square"></div>
<footer>Bottom</footer>
```

```
header{
  position: relative;
  top: 100px;
  background-color: #FF4136;
  height: 100px;
}

nav{
  position: absolute;
  background-color: #FFDC00;
  width: 50%;
  right: 0;
  bottom: 0;
}

.square{
  position: absolute;
  width: 50px;
  height: 50px;
}
```

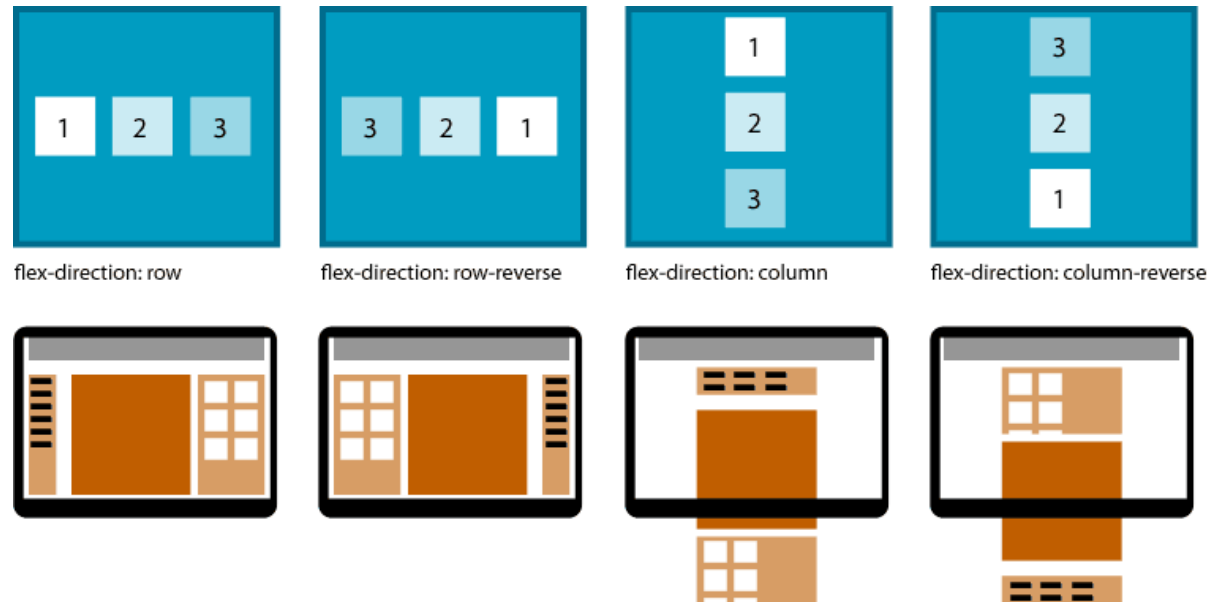
```
<div class="box">a</div>
<div class="box">b</div>
<div class="box">c</div>
<div class="box">d</div>
<h1>title</h1>
```

```
.box{
  border: 3px solid #FF851B;
  width: 30%;
  float:left;
  margin: 1%;
}

h1{
  clear: left;
}
```

CSS: Flexbox

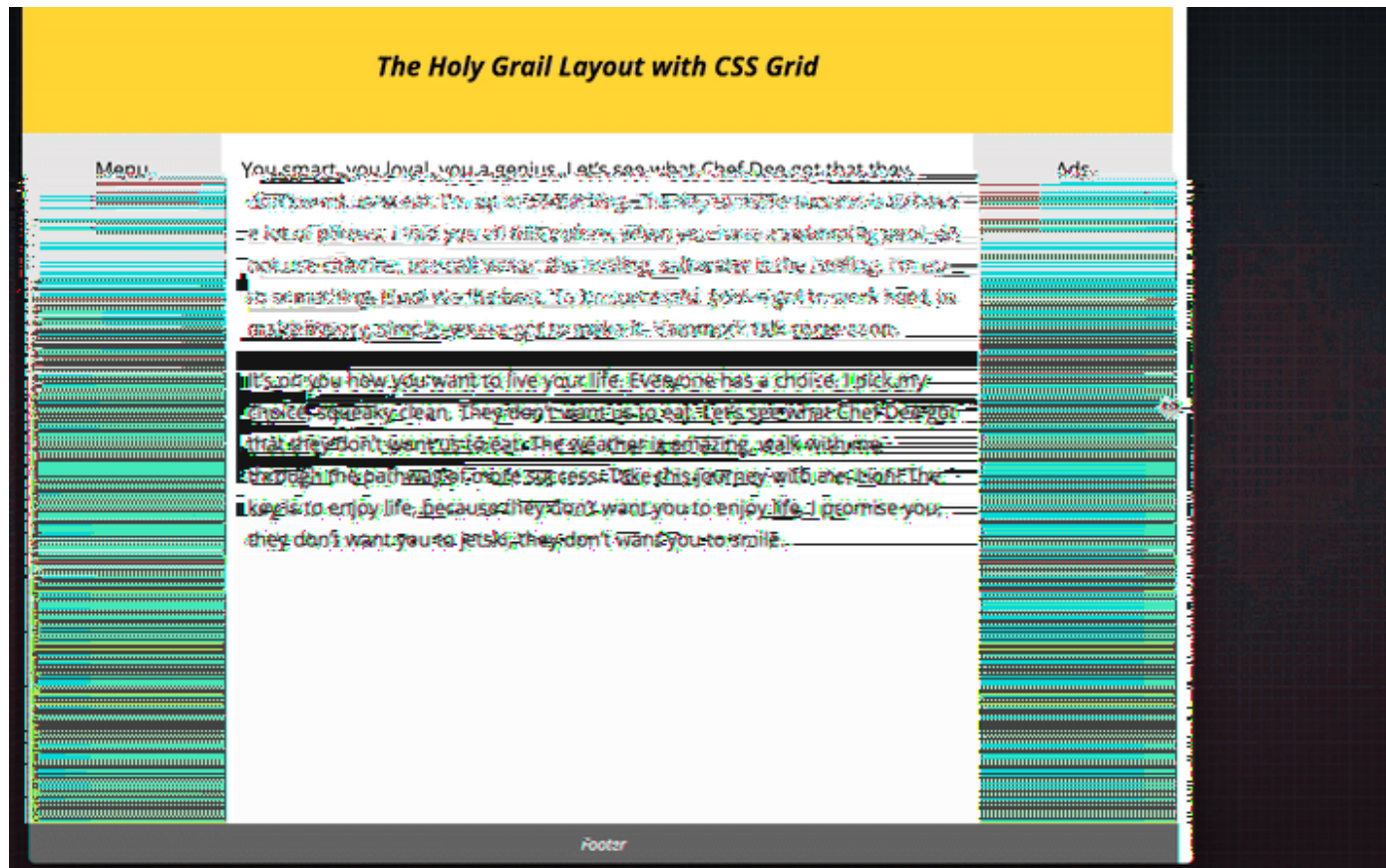
<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>



<https://philipwalton.github.io/solved-by-flexbox/>

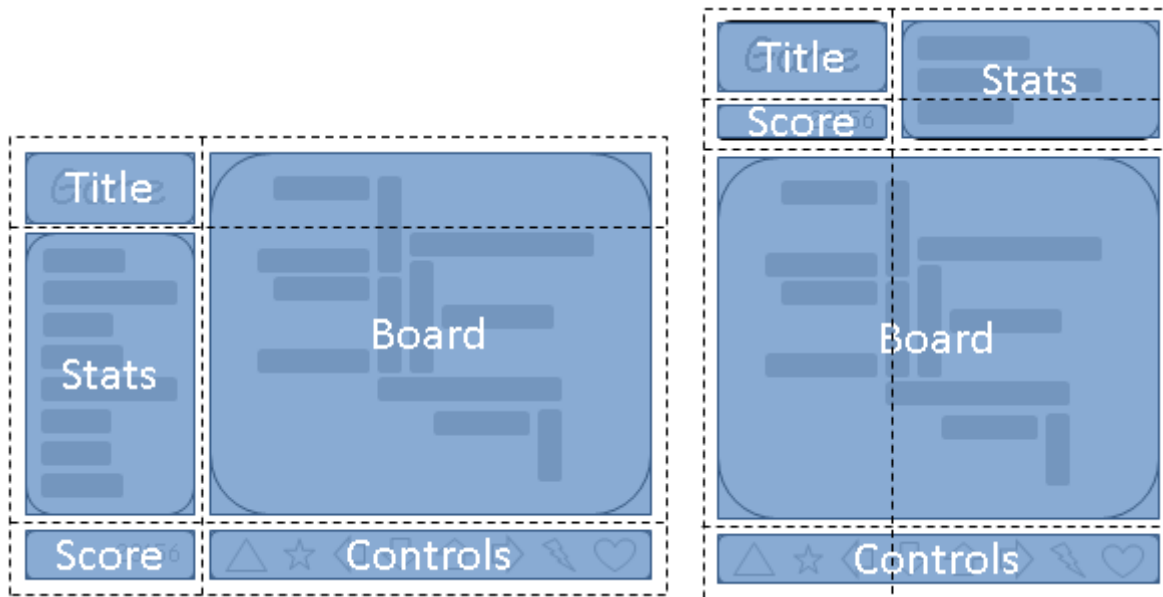
Exercise: Flexbox Froggy

CSS: Grid



Exercise: Grid Garden

Source-Order Independence



<https://drafts.csswg.org/css-grid/>

<https://gridbyexample.com/>

<https://css-tricks.com/snippets/css/complete-guide-grid/>

CSS: Media Queries

Lets the presentation of content be **tailored** to a specific range of output devices **without having to change the content** itself.

```
@media some rule [and/or] some other rule { CSS for only this selection }
```

rules: min-width, max-width, min-device-width, max-device-width, orientation: [landscape | portrait], ...

https://developer.mozilla.org/fr/docs/Web/CSS/Media_queries

CSS: Media Queries Bootstrap3 breakpoints

```
/* Custom, iPhone Retina */
@media only screen and (min-width : 320px) {

}
/* Extra Small Devices, Phones */
@media only screen and (min-width : 480px) {

}
/* Small Devices, Tablets */
@media only screen and (min-width : 768px) {

}
/* Medium Devices, Desktops */
@media only screen and (min-width : 992px) {

}
/* Large Devices, Wide Screens */
@media only screen and (min-width : 1200px) {

}
```

Media Queries Examples



```
<div id="box1">Box 1</div><div id="box2">Box 2</div><div id="box3">Box 3</div>
```

```
*{ box-sizing: border-box; }
body{ font-family: sans-serif; font-size: 20px; }
/* default mobile first */
#box1{ background-color: #3D9970; }
#box2{ background-color: #FF4136; }
#box3{ background-color: #FFDC00; }
#box1, #box2, #box3{ padding: 1em; }
/* first breakpoint */
@media only screen and (min-width : 600px) {
  #box1, #box2, #box3{
    float:left;
    width: 50%;
    text-align: center;
    border: 1em solid #fff;
  }
}
/* second breakpoint */
@media only screen and (min-width : 992px) {
  #box1, #box2, #box3{
    width: 33%;
    text-align: right;
  }
}
```

CSS Variables (Custom Properties)

Can be changed dynamically, follow CSS inheritance, can be changed from JavaScript

write:

```
--var_name: value;
```

read:

```
var(--var_name);
```

<https://drafts.csswg.org/css-variables/>

<https://www.smashingmagazine.com/2017/04/start-using-css-custom-properties/>

```
<h1>Title</h1>
<p>Some Text <span class="primary">primary text color</span></p>
<div class="box">box</div>
<div class="box">box</div>
```

```
:root { /* or any other css selector */
  --primary-color: red;
  --indent-size: 10px;
  --indent-x1: calc(2*var(--indent-size));
  --spacer: 10;
}

h1, h2 {
  color: var(--primary-color);
}

.primary {
  color: var(--primary-color);
}

.box {
  padding: var(--indent-x1);
}

.box2{
  padding: var(--spacer)px 0; /* DOESN'T work */
  padding: calc(var(--spacer)*1px) 0; /* WORKS */
}
```

CSS transition and animation



- <http://easings.net/> (easing functions)
- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Animations/Using_CSS_animations

References

- <http://caniuse.com/>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element>
- <https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formu>
- http://tympanus.net/codrops/css_reference/
- <https://developer.mozilla.org/fr/docs/Web/CSS/Reference>
- <http://www.w3.org/Style/css3-selectors-updates/WD-css3-selec>
- <http://css.sutterlity.fr/>
- <http://fr.learnlayout.com/>
- https://developer.mozilla.org/fr/docs/CSS/Premiers_pas



Sources

- Cours TWEB@heig-vd, Olivier Liechti
<https://github.com/wasadigi/Teaching-HEIGVD-TWEB/>
- yeoman pictures - <http://yeoman.io/>

