HYP TECHNOLOGY / FRONT-END

BOOTSTRAP

Mirko Gelsomini

Ways to get help:

0. Google or StackOverflow
1. After class – most immediate way
2. Via e-mail: mirko.gelsomini@polimi.it
3. private meeting at I3Lab, Via Golgi 39, building 21

iNNOVATIVE iNTERACTIVE iNTERFACES Laboratory
POLITECNICO MILANO 1863

Quiz #1

How do you select an element with id "demo"?

1. .demo
2. *demo
3. demo
4. #demo

Quiz #2

How do you select elements with class name "test"?

1. test
2. #test
3. *test
4. .test

Quiz #3

Which CSS property is used to change the text color of an element?

1. text-color
2. fgcolor
3. color

Quiz #4

How do you add a background color for all <h1> elements?

1. all.h1 {background-color:#FFFFFF;}
2. h1.all {background-color:#FFFFFF;}
3. h1 {background-color:#FFFFFF;}

Quiz #5

How do you display a border like this:
The top border = 10 pixels
The bottom border = 5 pixels
The left border = 20 pixels
The right border = 1pixel?

1. border-width:5px 20px 10px 1px;
2. border-width:10px 20px 5px 1px;
3. border-width:10px 1px 5px 20px;
4. border-width:10px 5px 20px 1px;

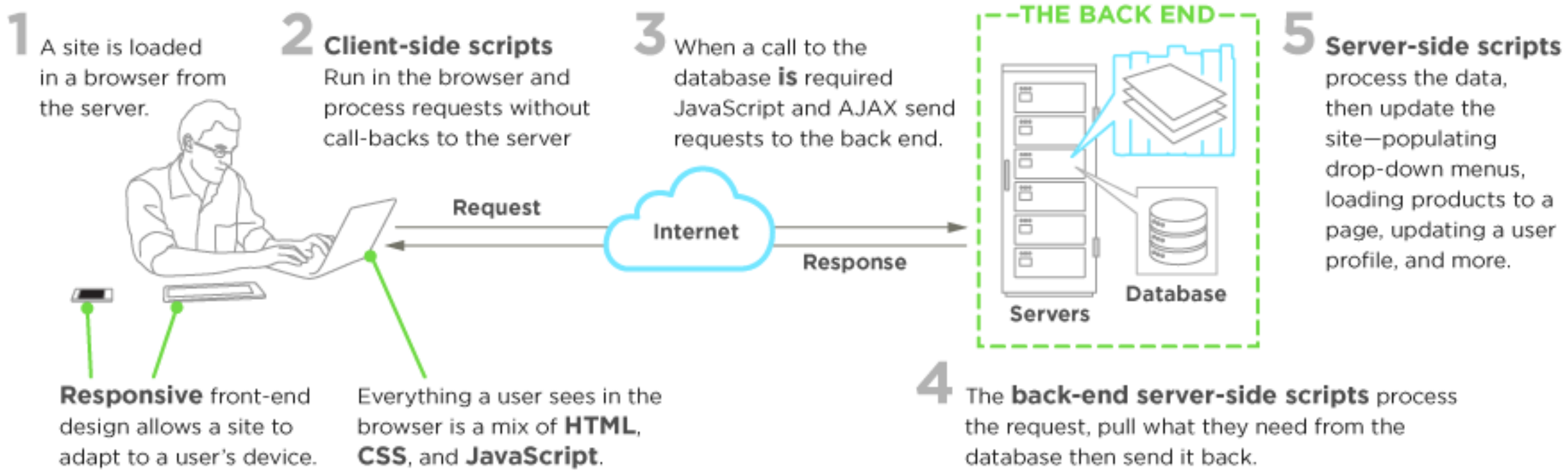
Quiz #6

How do you select all p elements inside a div element?

1. div p
2. div.p
3. p.div

Don’t know?: https://www.w3schools.com/cssref/css_selectors.asp

FRONT-END DEVELOPMENT




CSS
IS
AWESOME

Let’s solve the problem

HTML

```
<p> CSS IS
AWESOME</p>
```

CSS

```
p{
        width:30px;
        height:30px;
        border: 1px solid black;
        word-wrap: break-word;
}
```

EUROPEAN MEDICINES AGENCY

SCIENCE MEDICINES HEALTH

https://www.consilium.europa.eu/media/21805/amsterdam-ema-bidbook.pdf

https://www.consilium.europa.eu/media/21825/milan-ema-offer.pdf

HTML

HEAD

BODY

Structural Layer

HTML with CSS

APPEARANCE

LOOK

Presentation Layer

HTML, CSS & Bootstrap

Responsive,
& more stuff

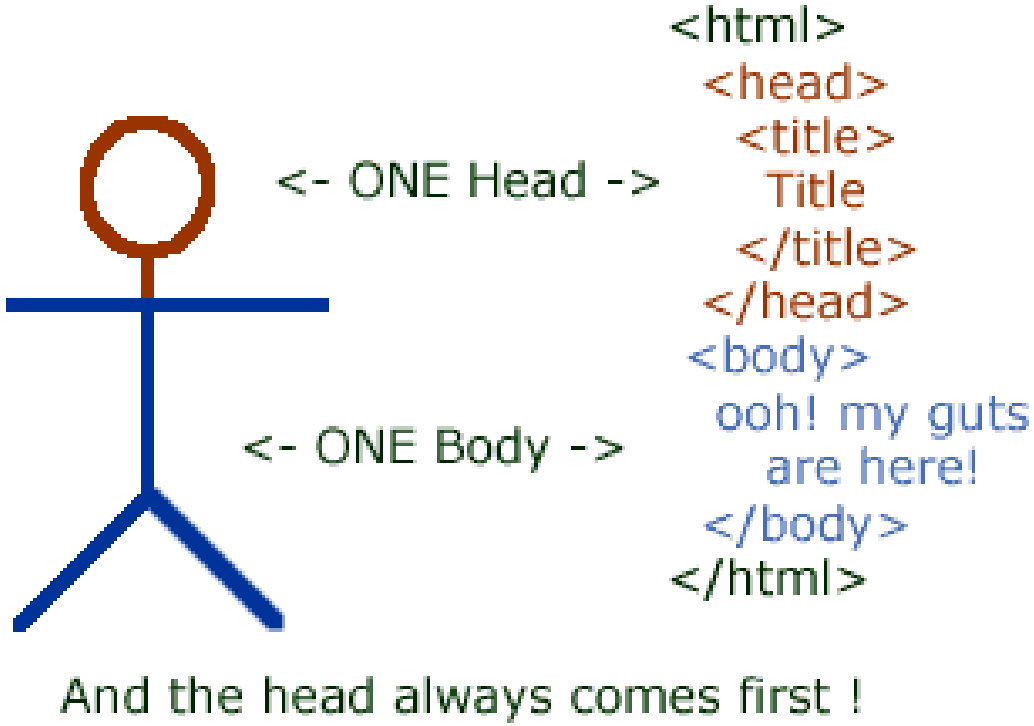
Action Layer

HTML, CSS, Bootstrap & JS

Does anyone need Caption?

HTML - HyperText Markup Language

- <!DOCTYPE html>
- <html>
  - <head>
  - </head>
  - <body>
  - </body>
- </html>



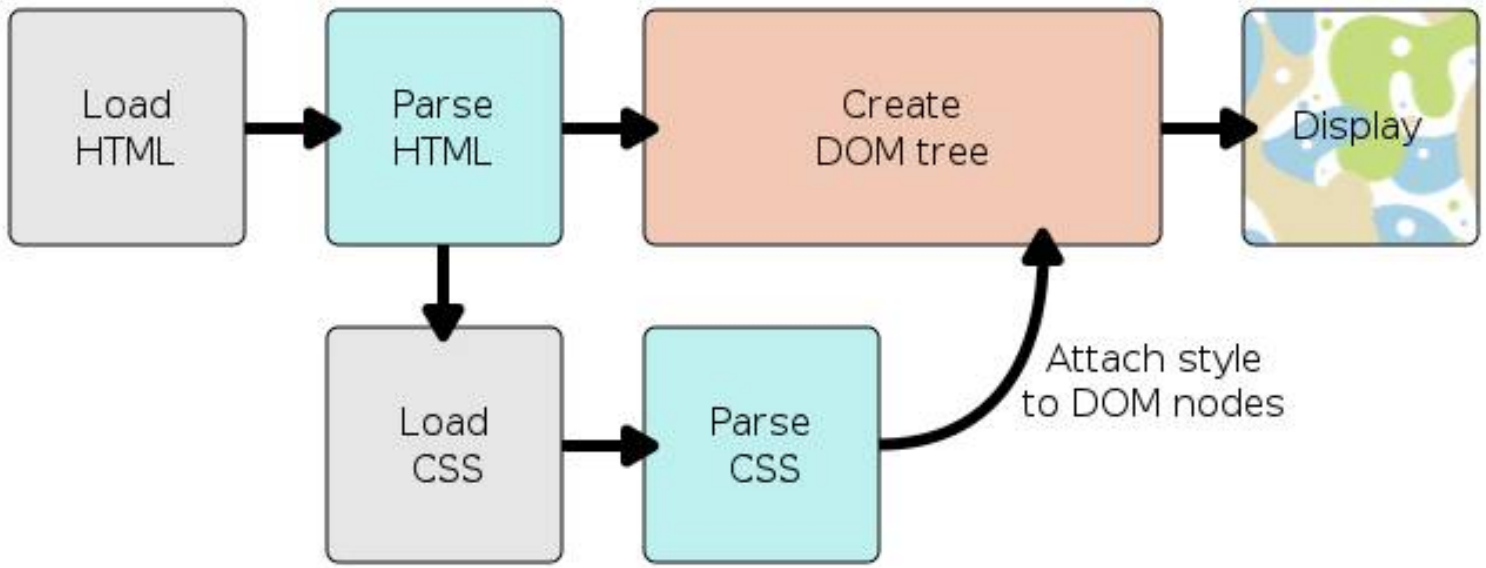
Connecting HTML, CSS and JS


```
<head>
    <link rel="stylesheet" type="text/css" href="assets/css/style.css">
</head>
```


```
<script src="assets/js/script.js"></script>
```

Browser’s job

- The browser converts HTML into the DOM (Document Object Model). The DOM represents the document in the computer's memory. It combines the document's content with its style.
- The browser displays the contents of the DOM.

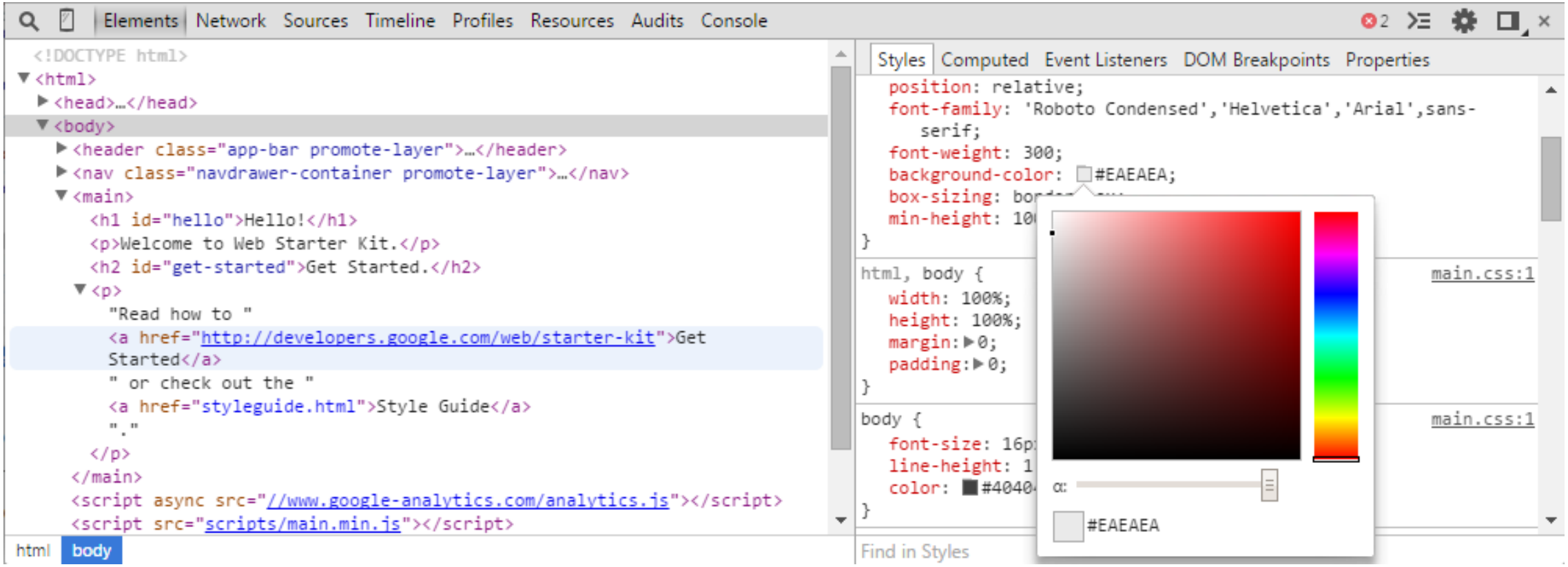


How DOM works

```
<p>
  Let's use:
  <span>Cascading</span>
  <span>Style</span>
  <span>Sheets</span>
</p>
```

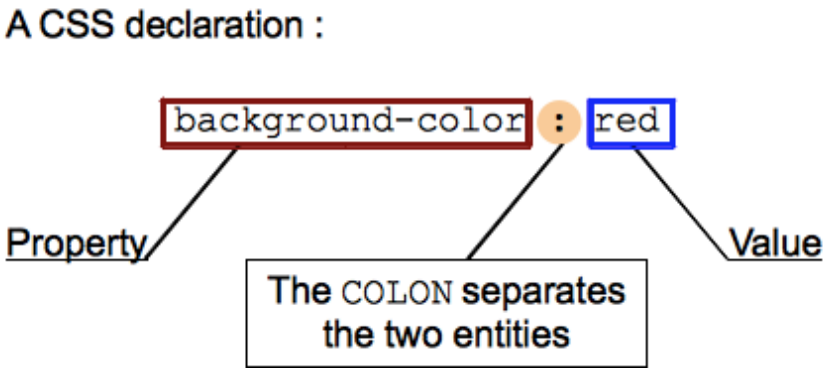
```
P
├─ "Let's use:"
├─ SPAN
|  └─ "Cascading"
├─ SPAN
|  └─ "Style"
└─ SPAN
   └─ "Sheets"
```

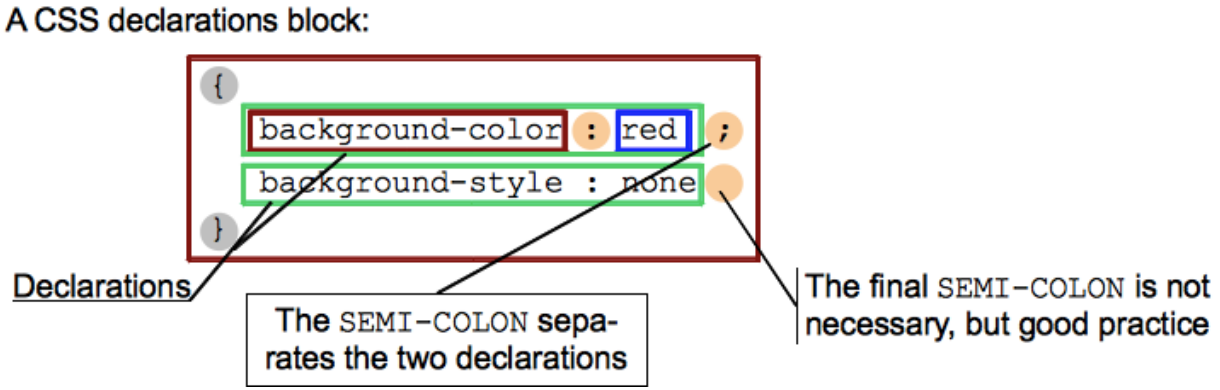
Dev Tools (F12)



https://developer.chrome.com/devtools

CSS Syntax





Comments in CSS begin with /* and end with */.

Important:
If a property is unknown or if a value is not valid for a given property, the declaration is deemed *invalid* and is wholly ignored by the browser's CSS engine.

In CSS (and other web standards), US spelling has been agreed on as the standard to stick to where uncertainty arises. For example, color (as seen in the above code) should always be spelt color. colour won't work.

HTML tags and CSS Selectors

```
<p>Hello Mickey!!</p>
```

```
p {
   color:red;
}
```

```
<p class="blue">Hello Pluto!!</p>
```

```
.blue{
   color:blue;
}
```

```
<p id="pluto">Hello Pippo!!</p>
```

```
#pluto{
   color:green;
}
```

Cascade – order matters!!!

- An element may be matched by several selectors, therefore several rules may set a given property multiple times. CSS defines which one has precedence over the others and must be applied: this is called the **cascade algorithm.**

- **Importance:** ID > class > element

- When importance is equal the last assignment commands!

More: media queries for printing

First of all, I want to print my website changing the view by removing/edit some contents and styles:

```
<link rel="stylesheet" href="print.css" media="print">
```

Or write inside your css:

```
@media print {
/* print style sheets go here */
}
```

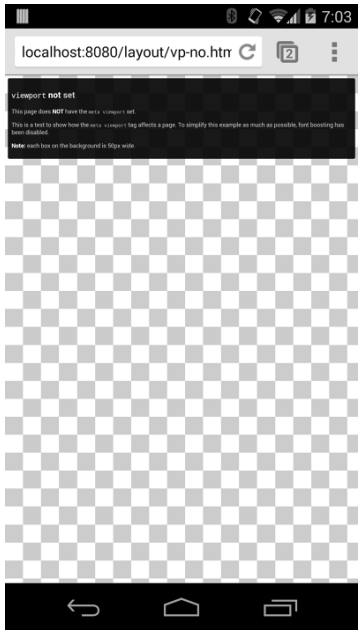
https://developers.google.com/web/fundamentals/layouts/rwd-fundamentals/use-media-queries?hl=en

Responsivity with css

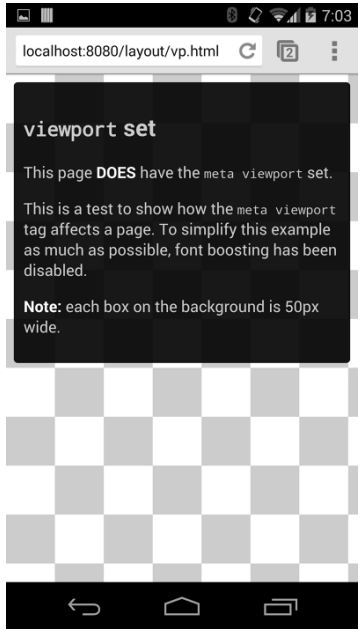
1) Set the viewport

- Use meta viewport tag to control the width and scaling of the browsers viewport.
- Include width=device-width to match the screen's width in device independent pixels.
- Include initial-scale=1 to establish a 1:1 relationship between CSS pixels and device independent pixels.
- Ensure your page is accessible by not disabling user scaling.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

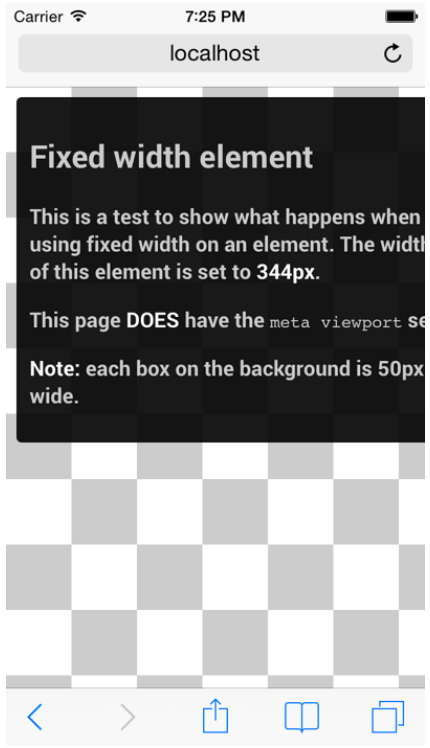


Using the meta viewport value `width=device-width` instructs the page to match the screen's width in device independent pixels. This allows the page to reflow content to match different screen sizes, whether rendered on a small mobile phone or a large desktop monitor.



2) Size contents

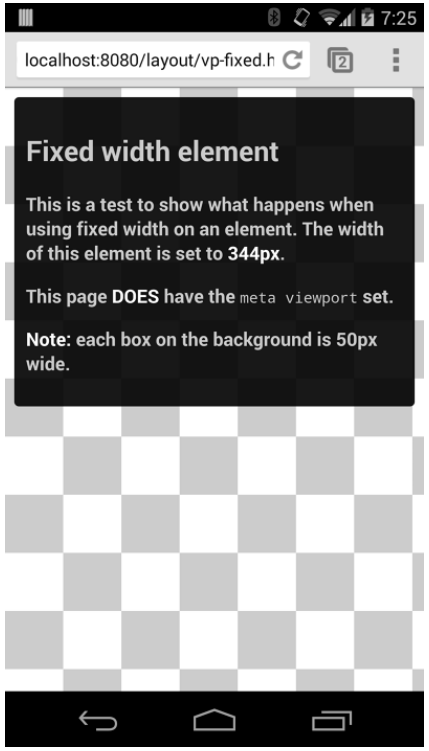
- Do not use large fixed width elements.
- Content should not rely on a particular viewport width to render well.
- Use CSS media queries to apply different styling for small and large screens.



Setting large absolute CSS widths for page elements (such as the example on the left), will cause the `div` to be too wide for the viewport on a narrower device (e.g. a device with a width of 320 CSS pixels, such as an iPhone).

Instead, consider using relative width values, such as **`width: 100%`**

Similarly, beware of using large absolute positioning values that may cause the element to fall outside the viewport on small screens.



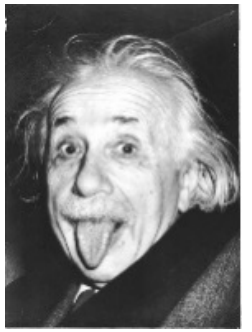
3) Use media queries

Media queries enable us to create a responsive experience, where specific styles are applied to small screens, large screens and anywhere in between. The media query syntax allows for the creation of rules that can be applied depending on device characteristics.

```
@media (query) {
/* CSS Rules used when query matches */
}
```

attribute	Result
`min-width`	Rules applied for any browser width over the value defined in the query.
`max-width`	Rules applied for any browser width below the value defined in the query.
`min-height`	Rules applied for any browser height over the value defined in the query.
`max-height`	Rules applied for any browser height below the value defined in the query.
`orientation=portrait`	Rules applied for any browser where the height is greater than or equal to the width.
`orientation=landscape`	Rules for any browser where the width is greater than the height.

4) Be relative!



- A key concept behind responsive design is fluidity and proportionality as opposed to fixed width layouts. Using relative units for measurements can help simplify layouts and prevent accidentally creating components that are too big for the viewport.
- For example, setting width: 100% on a top level div, ensures that it spans the width of the viewport and is never too big or too small for the viewport. The div will fit, no matter if it's a 320px wide iPhone, 342px wide Blackberry Z10 or a 360px wide Nexus 5.
- In addition, using relative units allows browsers to render the content based on the users zoom level without the need for adding horizontal scroll bars to the page.

NO

```
div.fullWidth {
  width: 320px;
  margin-left: auto;
  margin-right: auto;
}
```

YES

```
div.fullWidth {
  width: 100%;
}
```

Tryout: http://googlesamples.github.io/web-fundamentals/samples/layouts/rwd-fundamentals/media-queries.html

5) Standardize

https://css-tricks.com/snippets/css/media-queries-for-standard-devices/

```
/* Smartphones (portrait and landscape) ----------- */
@media only screen and (min-device-width : 320px) and (max-device-width : 480px) { /* Styles */ }

/* Smartphones (landscape) ----------- */
@media only screen and (min-width : 321px) { /* Styles */ }

/* Smartphones (portrait) ----------- */
@media only screen and (max-width : 320px) { /* Styles */ }

/* iPads (portrait and landscape) ----------- */
@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) { /* Styles */ }

/* iPads (landscape) ----------- */
@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : landscape) { /* Styles */ }

/* iPads (portrait) ----------- */
@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : portrait) { /* Styles */ }

/* Desktops and laptops ----------- */
@media only screen and (min-width : 1224px) { /* Styles */ }

/* Large screens ----------- */
@media only screen and (min-width : 1824px) { /* Styles */ }

/* iPhone 4 ----------- */
@media only screen and (-webkit-min-device-pixel-ratio : 1.5), only screen and (min-device-pixel-ratio : 1.5) { /* Styles */ }
```

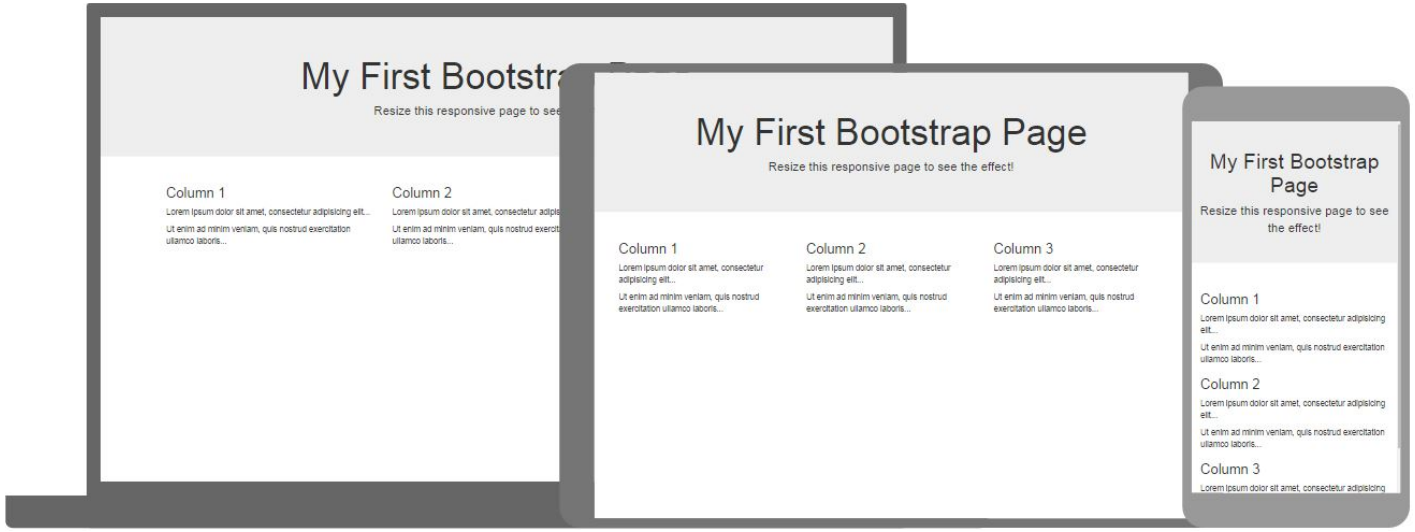
width refers to the width of the rendering area, **device-width** refers to the entire screen area

Bootstrap

- **Bootstrap** is an HTML5 & CSS3 framework designed to help you kickstart the development of web-apps and sites.
- E.g. https://wrapbootstrap.com/ or https://startbootstrap.com/ are marketplaces for premium **Bootstrap themes** and **templates**.
- Impress your clients and visitors while using a single, rock-solid foundation.
- Some other links:

https://shapebootstrap.net/free-templates, https://bootswatch.com/, https://bootstrapmade.com/, https://almsaeedstudio.com/blog/10-Free-Responsive-Bootstrap-Templates-For-2016

Boostrap



Why Use Bootstrap? Advantages of Bootstrap:

- **Easy to use:** Anybody with just basic knowledge of HTML and CSS can start using Bootstrap
- **Responsive features:** Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
- **Mobile-first approach:** In Bootstrap 3, mobile-first styles are part of the core framework
- **Browser compatibility:** Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Safari, and Opera)

Link to Bootstrap

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/boo
tstrap.min.css">

<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></scr
ipt>

<!-- Latest compiled JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></
script>
```

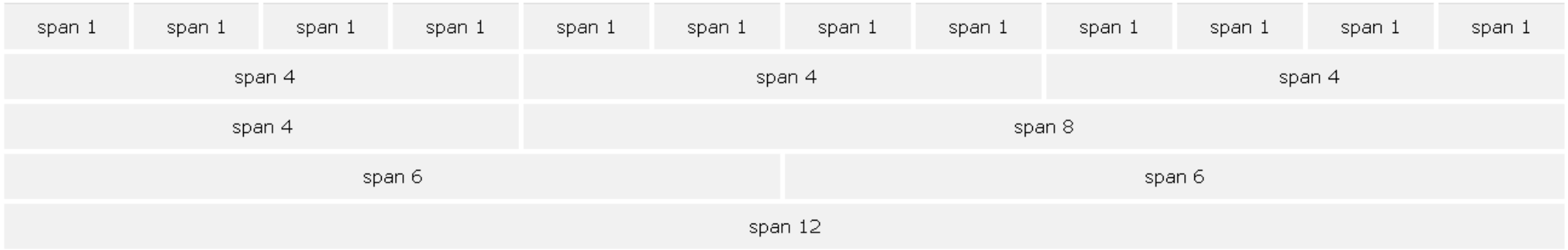
Precompiled Bootstrap

Once downloaded, unzip the compressed folder to see the structure of (the compiled) Bootstrap. You'll see something like this:

```
bootstrap/
├── css/
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap.min.css.map
│   ├── bootstrap-theme.css
│   ├── bootstrap-theme.css.map
│   ├── bootstrap-theme.min.css
│   └── bootstrap-theme.min.css.map
├── js/
│   ├── bootstrap.js
│   └── bootstrap.min.js
└── fonts/
    ├── glyphicons-halflings-regular.eot
    ├── glyphicons-halflings-regular.svg
    ├── glyphicons-halflings-regular.ttf
    ├── glyphicons-halflings-regular.woff
    └── glyphicons-halflings-regular.woff2
```

This is the most basic form of Bootstrap: precompiled files for quick drop-in usage in nearly any web project. We provide compiled CSS and JS (`bootstrap.*`), as well as compiled and minified CSS and JS (`bootstrap.min.*`). CSS source maps (`bootstrap.*.map`) are available for use with certain browsers' developer tools. Fonts from Glyphicons are included, as is the optional Bootstrap theme.

Bootstrap Grid System



The Bootstrap grid system has four classes:

- xs (for phones)
- sm (for tablets)
- md (for desktops)
- lg (for larger desktops)

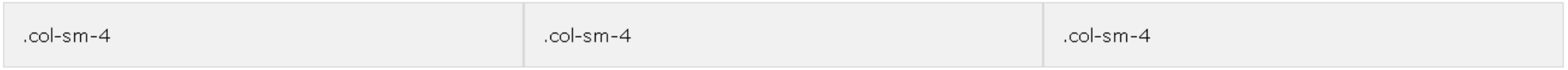
Basic Structure of a Bootstrap Grid

```
<div class="container">
    <div class="row">
      <div class="col-x-y"></div>
    </div>
    <div class="row">
      <div class="col-x-y"></div>
      <div class="col-x-y"></div>
      <div class="col-x-y"></div>
    </div>
</div>
```

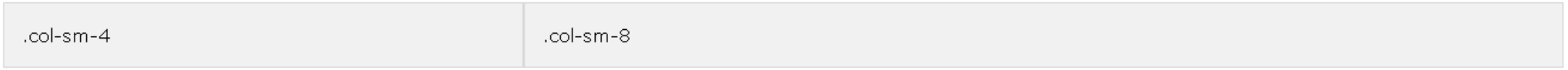
x can be any value between xs, sm, md, lg

y can be any value from 1 to 12 included

Example: Equal and Unequal Columns




```
<div class="row">
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-4">.col-sm-4</div>
</div>
```




```
<div class="row">
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-8">.col-sm-8</div>
</div>
```

Basic template

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <title>Page Title</title>

   <!-- Bootstrap -->
   <link href="assets/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>
   <h1>Hello, world!</h1>

   <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
   <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
   <!-- Include all compiled plugins (below), or include individual files as needed -->
   <script src="assets/js/bootstrap.min.js"></script>
</body>
```

Grid options

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	`.col-`	`.col-sm-`	`.col-md-`	`.col-lg-`	`.col-xl-`
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

Bootstrap Docs

Old version: http://getbootstrap.com/2.3.2/getting-started.html

Latest version: http://getbootstrap.com/getting-started/

On the web – in case you missed this class

- https://developers.google.com/web/fundamentals/ (en - free)
- http://www.codecademy.com/tracks/web (en - free)
- http://www.html.it/guide/guida-html5/ (ita - free)
- https://www.codeschool.com/paths/html-css (en)
- http://teamtreehouse.com/tracks/front-end-web-development (en)

- Bootstrap: https://www.tutorialspoint.com/bootstrap/index.htm

- Official Ref (en):
  - HTML: http://www.w3schools.com/html/
  - CSS: http://www.w3schools.com/css/

Being stuck?

0. Google (Stack Overflow suggested)
1. Meet Mirko just After class – most immediate way
2. Via e-mail: mirko.gelsomini@polimi.it
3. Meet Mirko at I3Lab, Via Golgi 39, building 21