HYP TECHNOLOGY / FRONT-END

CSS

Mirko Gelsomini

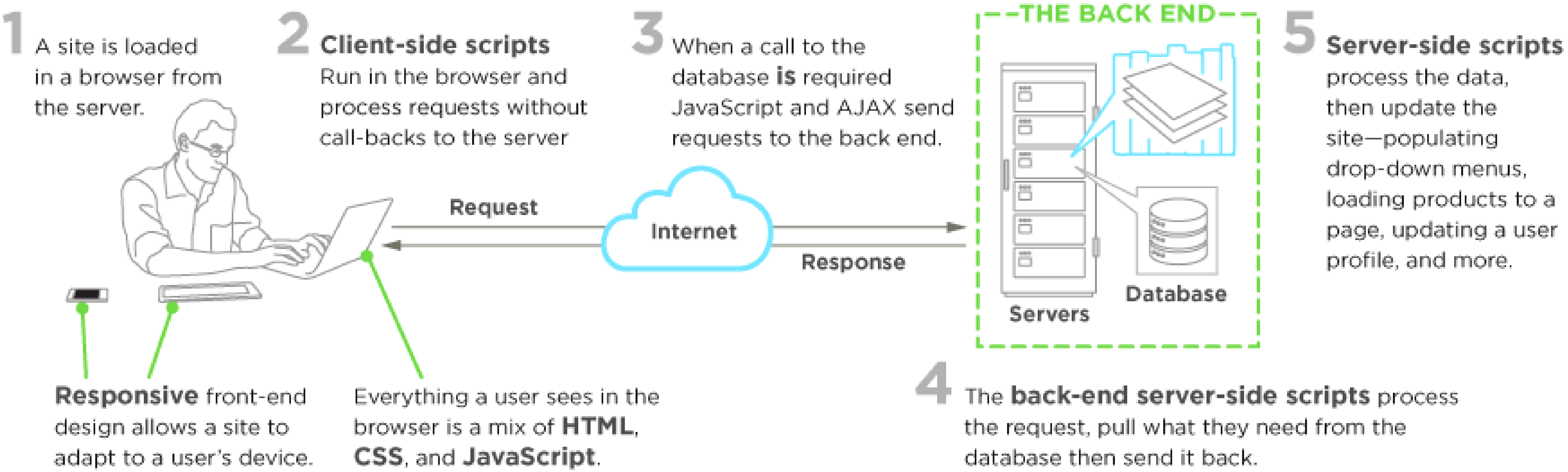
Ways to get help:

0. Google
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

FRONT-END DEVELOPMENT



1 A site is loaded in a browser from the server.

2 **Client-side scripts** Run in the browser and process requests without call-backs to the server

3 When a call to the database **is** required JavaScript and AJAX send requests to the back end.

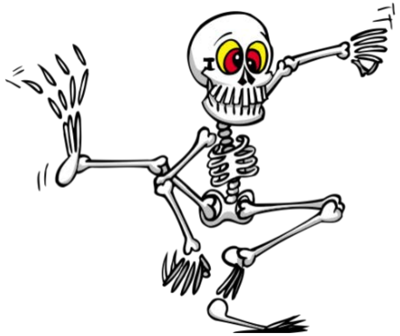
4 The **back-end server-side scripts** process the request, pull what they need from the database then send it back.

5 **Server-side scripts** process the data, then update the site—populating drop-down menus, loading products to a page, updating a user profile, and more.

Responsive front-end design allows a site to adapt to a user's device.

Everything a user sees in the browser is a mix of **HTML**, **CSS**, and **JavaScript**.

HTML



Content/Structure

CSS

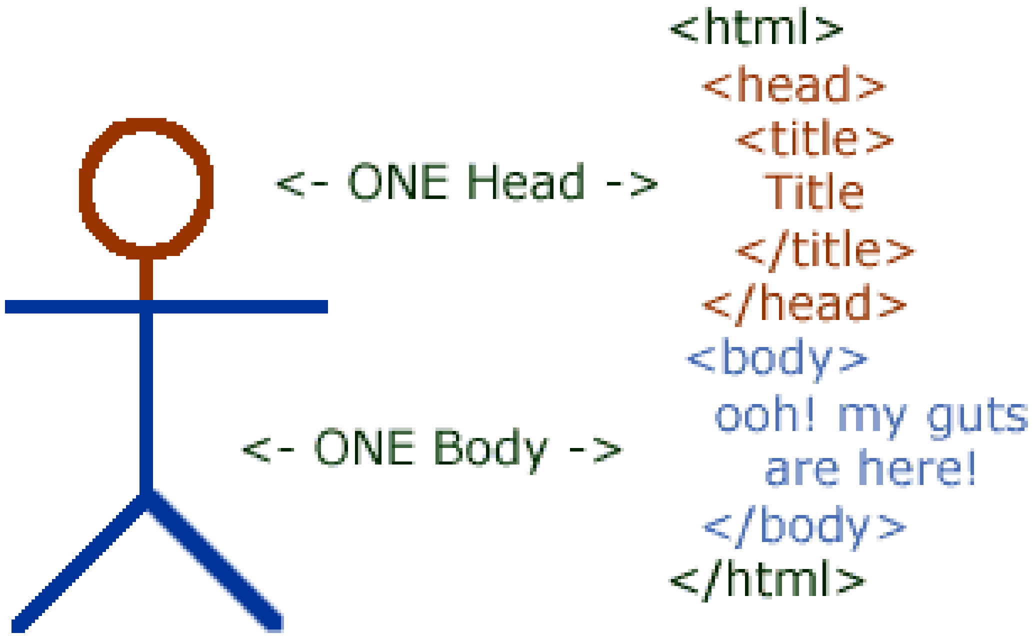
Style

JS

Dinamicity and Control

HTML - HyperText Markup Language

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



```
<html>
  <head>
    <title>
     Title
    </title>
  </head>
 <body>
   ooh! my guts
      are here!
  </body>
</html>
```

And the head always comes first !

HTML - index.html/htm

The browser will always look for a file called «index.*»

except you change the .htaccess (more in next classes)

Using a Default File Name Like **index.html** is a Security Feature as Well. ... If you don't put in an **index.html** file in a directory, by default most web servers will display a file listing of all the files in that directory.

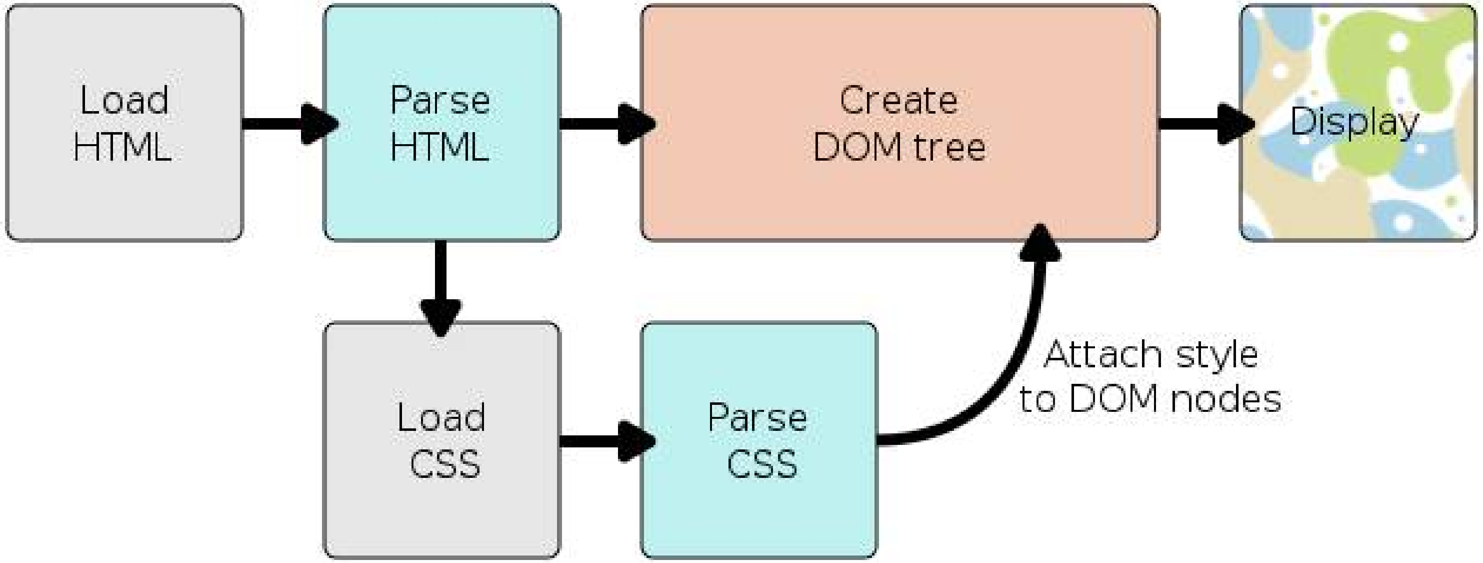
How DOM works

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

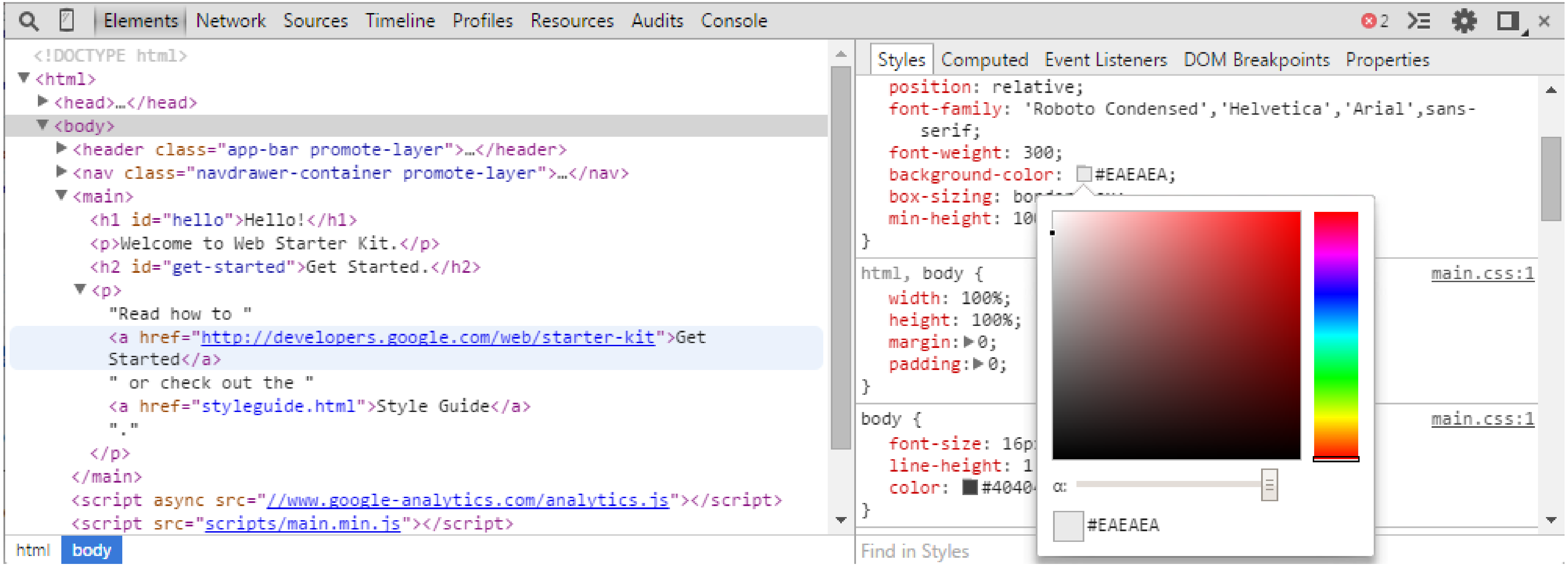
```
P
├─ "Let's use:"
├─ SPAN
|  └─ "Cascading"
├─ SPAN
|  └─ "Style"
└─ SPAN
   └─ "Sheets"
```

Browser’s job

- The browser converts HTML and CSS 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.



Dev Tools (F12)



```
<!DOCTYPE html>
▼<html>
  ▶<head>…</head>
  ▼<body>
    ▶<header class="app-bar promote-layer">…</header>
    ▶<nav class="navdrawer-container promote-layer">…</nav>
    ▼<main>
        <h1 id="hello">Hello!</h1>
        <p>Welcome to Web Starter Kit.</p>
        <h2 id="get-started">Get Started.</h2>
      ▼<p>
          "Read how to "
          <a href="http://developers.google.com/web/starter-kit">Get
          Started</a>
          " or check out the "
          <a href="styleguide.html">Style Guide</a>
          "."
        </p>
      </main>
      <script async src="//www.google-analytics.com/analytics.js"></script>
      <script src="scripts/main.min.js"></script>
```

```
  position: relative;
  font-family: 'Roboto Condensed','Helvetica','Arial',sans-
     serif;
  font-weight: 300;
  background-color: □#EAEAEA;
  box-sizing: bor
  min-height: 10
}
html, body {                                             main.css:1
  width: 100%;
  height: 100%;
  margin:▶0;
  padding:▶0;
}
body {                                                   main.css:1
  font-size: 16p
  line-height: 1
  color: ■#4040
}
```

https://developer.chrome.com/devtools

What is CSS?

CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once.

One structure, different styles

http://www.csszengarden.com/218/
http://www.csszengarden.com/219/
http://www.csszengarden.com/220/
…
http://www.mezzoblue.com/zengarden/alldesigns/

Connecting HTML, CSS and JS

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

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

Relative Url

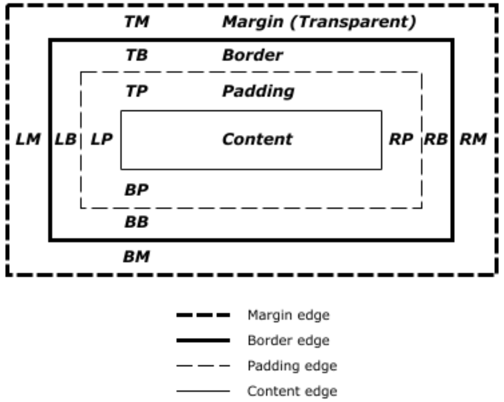
- To go down one folder, simple type the folder before the file.
  Example: **folder/file.txt** or **./folder/file.txt**

- To go up one folder, type the file name with “../“.
  Example: **../folder/file.txt**

Hands On CSS

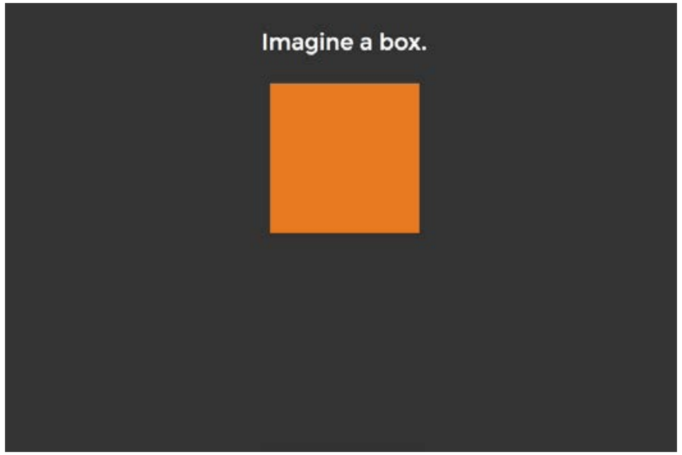
http://www.cheetyr.com/css-selectors

- Tags, Id, Classes, parenthood
- Pseudo-classes and multiselectors
- Positioning (margin, borders and padding)
- Floating, absolute, relative and fixed
- Dimension: width, height
- Font, Colors



Cool CSS Creations

- http://codepen.io/rgg/pen/rVgBEL
- http://codepen.io/rgg/pen/QbRyOq
- http://codepen.io/teles/pen/gbKeLR
- http://codepen.io/jcoulterdesign/pen/oXqZKZ
- http://codepen.io/HugoGroutel/pen/EVKdrK
- http://codepen.io/caraujo/pen/QbZbpg
- http://codepen.io/soulwire/pen/foktm
- http://codepen.io/TimPietrusky/pen/Bsegb
- http://codepen.io/jackrugile/pen/acAgx
- https://codepen.io/gxash/pen/YqmxWg

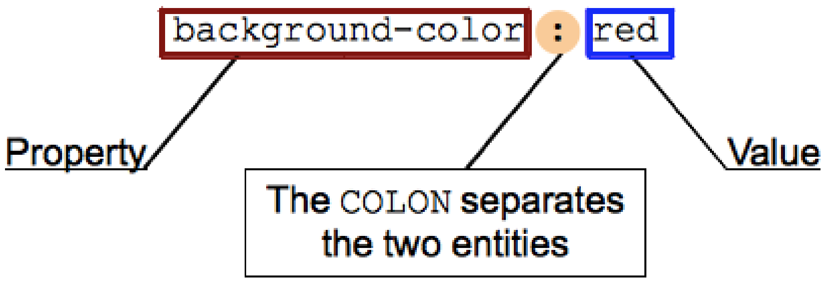


CSS – Reference (Cheat Sheet)

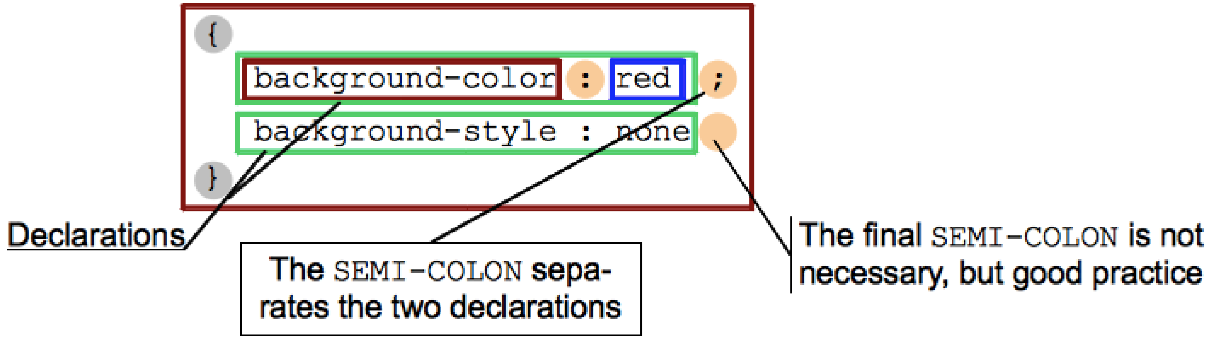
https://www.smashingmagazine.com/wp-content/uploads/images/css3-cheat-sheet/css3-cheat-sheet.pdf

CSS Syntax

A CSS declaration :



A CSS declarations block:


{
background-color
:
red
;
background-style : none
}
Declarations
The SEMI-COLON separates the two declarations
The final SEMI-COLON is not necessary, but good practice

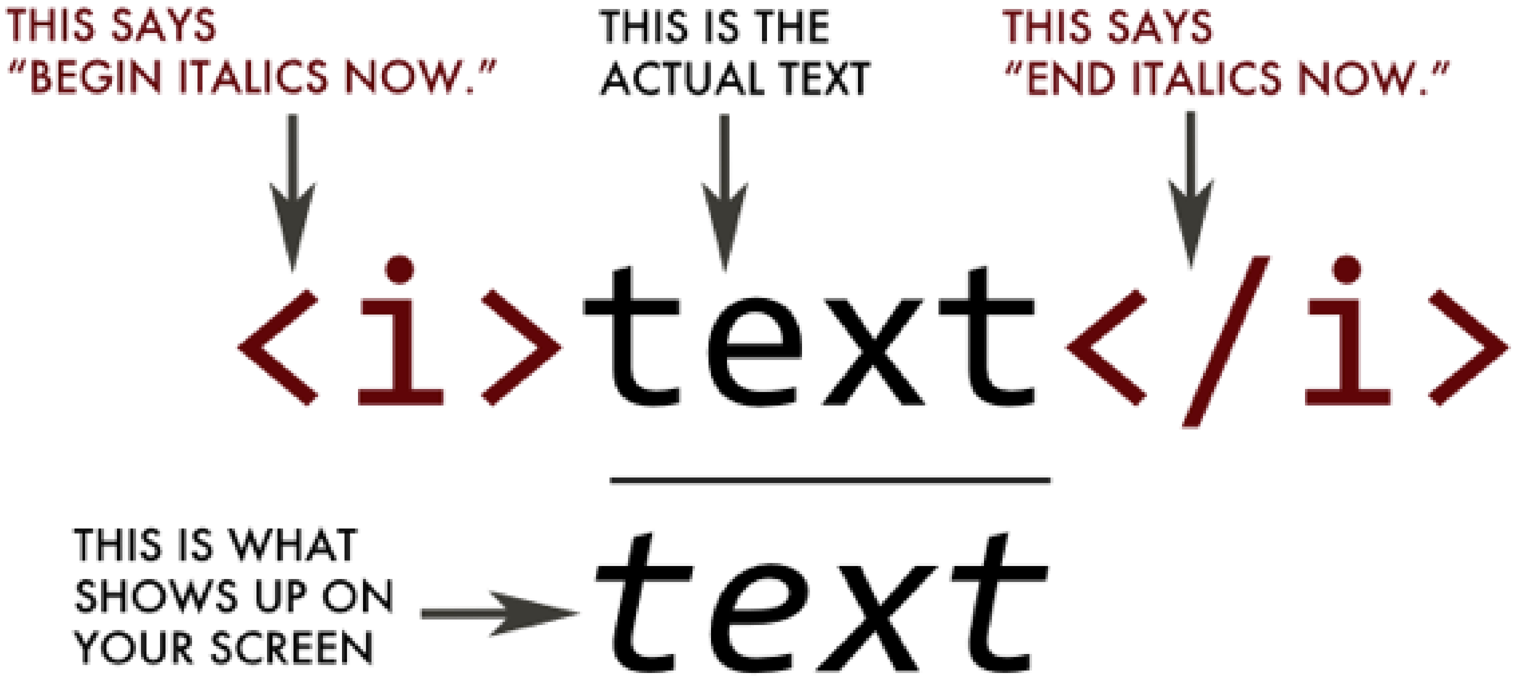
Comments in CSS begin with /* and end with */.

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

```
<tag attribute="value"> Content </tag>
```



THIS SAYS "BEGIN ITALICS NOW."
THIS IS THE ACTUAL TEXT
THIS SAYS "END ITALICS NOW."
<i>text</i>
THIS IS WHAT SHOWS UP ON YOUR SCREEN
text

HTML tags and CSS Selectors

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

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

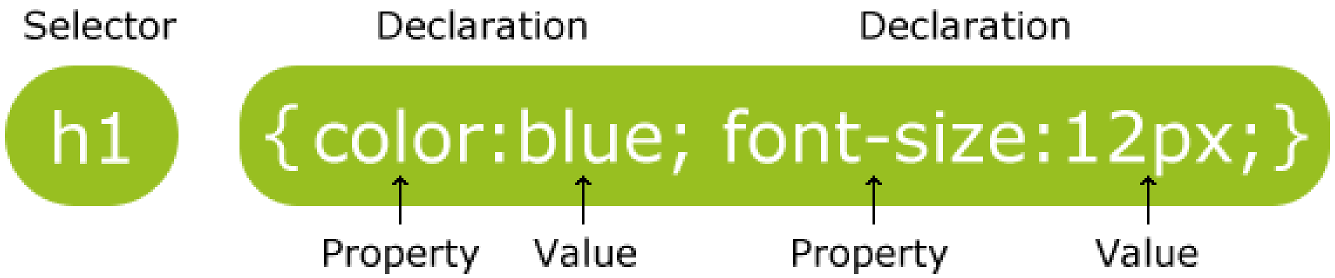
```
<p class="blue">Hello Pluto!!</p>
```

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

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

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

CSS Syntax and Selectors



Selector Declaration Declaration

h1 {color:blue; font-size:12px;}

Property Value Property Value

```
h1 {
    text-align: center;
    color: red;
}

h2 {
    text-align: center;
    color: red;
}

p {
    text-align: center;
    color: red;
}
```

Grouping

```
h1, h2, p {
    text-align: center;
    color: red;
}
```

Cascade – order matters!!!

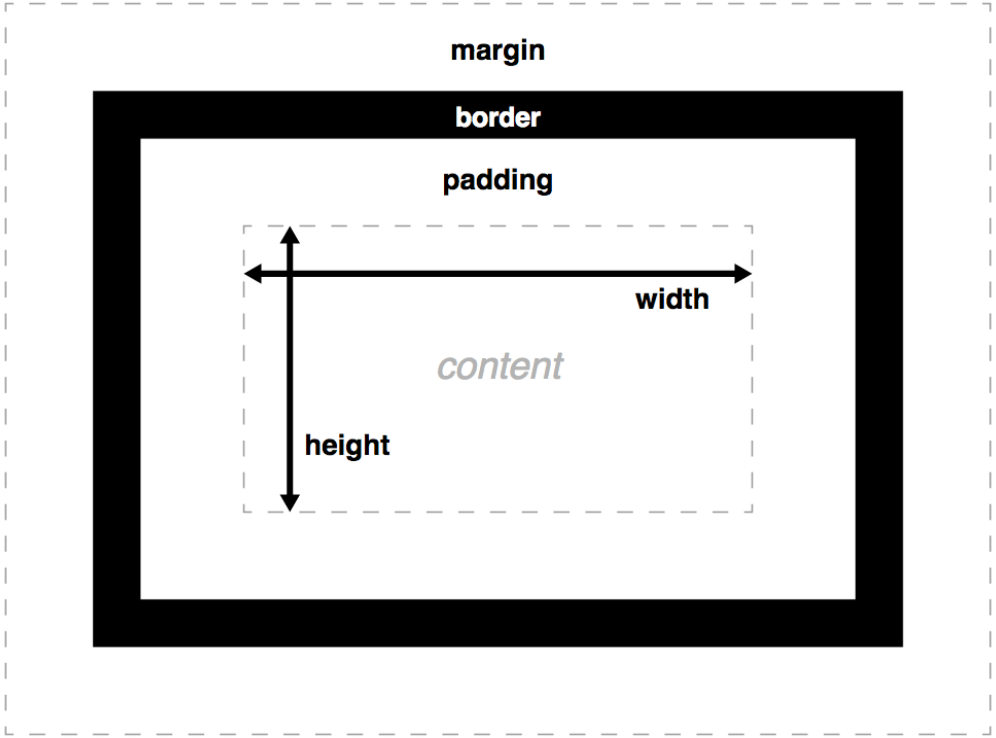
- An element may be matched by several selectors, therefore several rules may set a given property multiple times. CSS defines which one has precedence over the others and must be applied: this is called the **cascade algorithm.**

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

- When importance is equal the last assignment commands!

Box properties

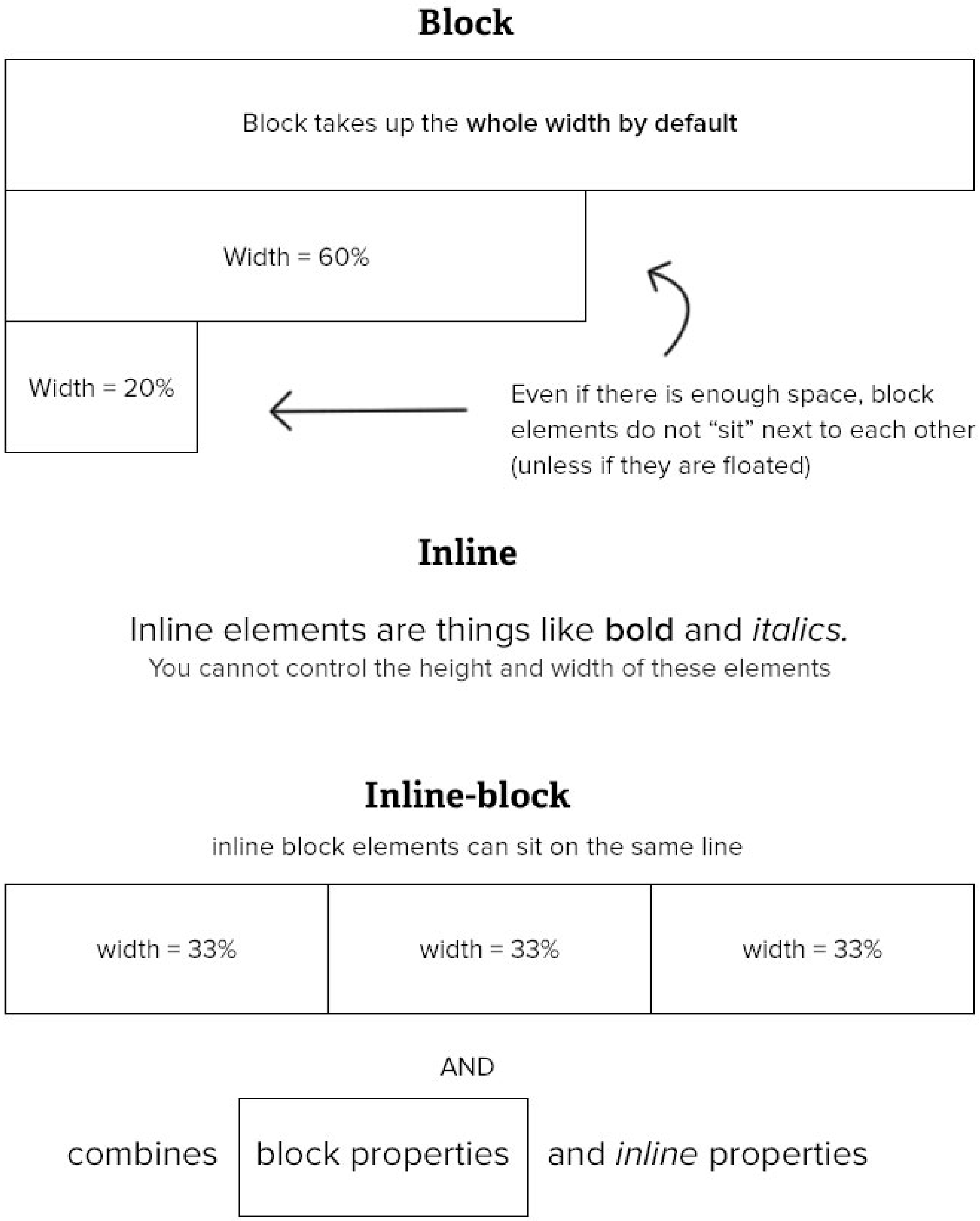
- Every element within a document is structured as a rectangular box inside the document layout, the size and "onion layers" of which can be tweaked using some specific CSS properties. The relevant properties are as follows:



Blocks

One of them:

```
div {
        display: block;
        display: inline-block;
        display: inline;
        display: none;
}
```



Block

Block takes up the **whole width by default**

Width = 60%

Width = 20%

Even if there is enough space, block elements do not “sit” next to each other (unless if they are floated)

Inline

Inline elements are things like **bold** and *italics.*

You cannot control the height and width of these elements

Inline-block

inline block elements can sit on the same line

width = 33%

width = 33%

width = 33%

AND

combines block properties and *inline* properties

Width and Height

- Width and height properties are used closely with display:block and display:inline to set the width and height of HTML elements while creating a website. Common units units for Width and Height are:
- **px** - Pixels.
- **em** - A unit of measurement, where 1 em = current font size.
- **%** - Percentages.
- **auto** - auto adaptation to width/height

Margins

```
div{
   margin-top: 20px;
   margin-bottom: 20px;
   margin-right: 10px;
   margin-left: 10px
}
```

Or

```
margin: 20px 10px 20px 10px;
```

margins with auto on the left and right are used to center an element with a display value of block.

```
div {
        margin: 0 auto;
}
```

Borders

- **border-width** – width of the border. Same units as width and height
- **border-style** – style of the border. Usual values are solid and dashed. For a complete list, take a look at W3 Schools Website
- **border-color** – color of the border. Hex, and rgb values can be used.

Or all together:

```
div{
	border: 1px solid black;
}
```

Float

- to position the edge of the targeted HTML content at the edge of one side of the parent container.
- Floats have 3 basic properties that you might use often:
  - **left**
  - **right**
  - **none** - removes the float

Background Color and Colors

- Background refers to the background of the HTML element. Like many CSS properties, background has a shorthand to it as well.

- **background-color**: color of the background. Takes #hex value or an rgb value
- **background-image**: url(URI). Takes on the path to your image.
- **background-repeat**: whether you would like the background to repeat if the width exceeds the background size. Other values are repeat, repeat-x and repeat-y.
- **background-position**: position of the background relative to the HTML element. Two values are needed here, X and Y, where X is the amount of offset from the left and Y is the amount of offset from the top. Takes on either unit values (as with width and height) or left,center,right and top,center,bottom for left and right respectively.

Font

- Fonts in general refer to the appearance of text in your website. There are a few things to look out for. Like other properties, font has a shorthand. Do note that it is common to see the font shorthand declaration only once in the whole CSS file. It is common to use the different properties at other times:

```
body {
	font: italic bold 20px "Proxima Nova", helvetica, arial, sans-serif;
}
```

Fonts

- **font-style**: Style of the font. valid values are either italic or normal. Defaults to normal. Optional property in font shorthand
- **font-variant:** variant of the font. valid values are normal and small caps. Defaults to normal. Optional property in font shorthand and is not often used
- **font-weight:** weight of font. determines if text is bold. valid values are normal, bold, bolder, or 100 - 900. Optional property in font shorthand
- **font-size:** size of font. Takes a value thats the same as width and height
- **line-height:** determines the amount of space above and below the text. Very important to ensure good readability. Takes on the same values as font, and also a unitless value. If a unitless value is used, it means the line height is a multiple of the value provided.
- **font-family:** area to declare typefaces and fontstack that you would like to use.

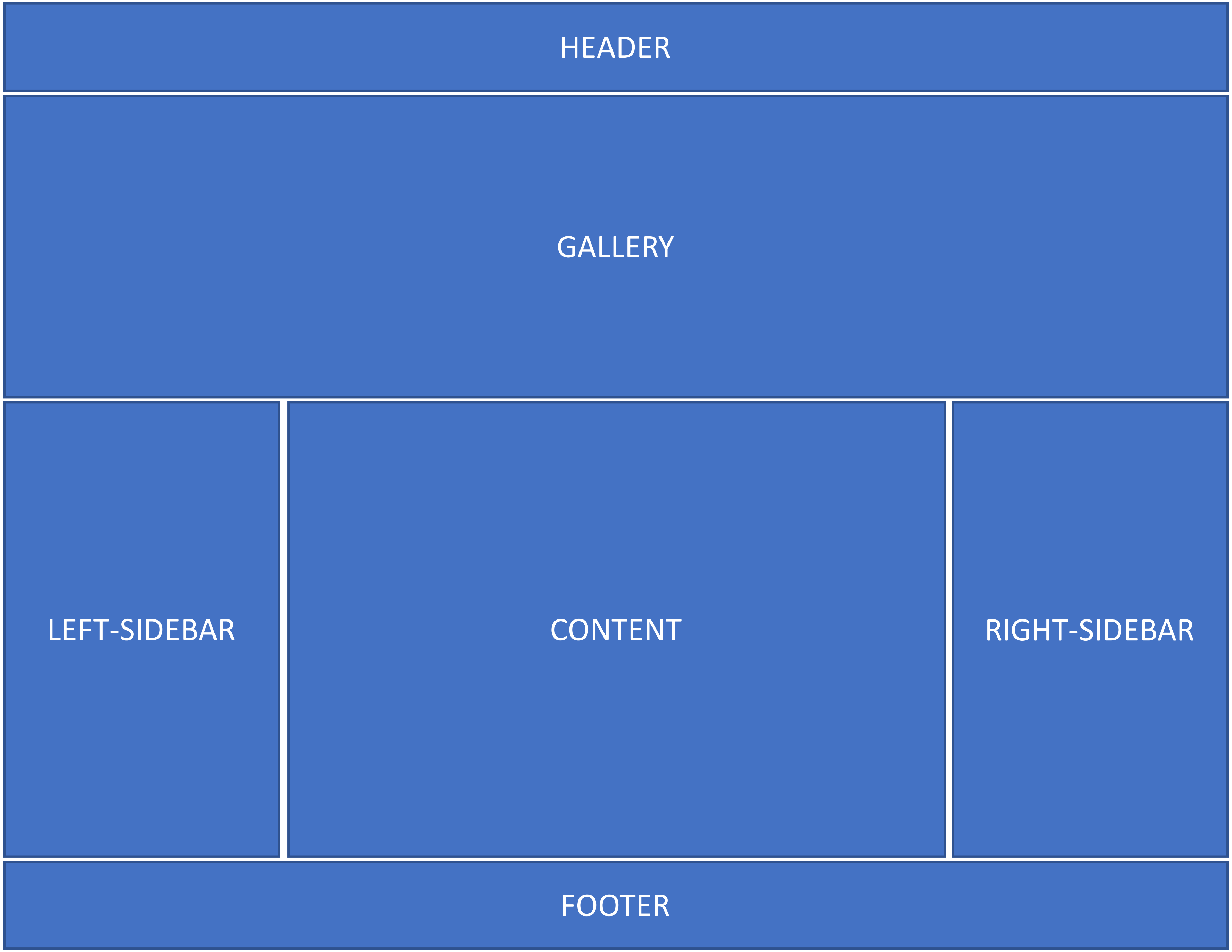
Un-common font styling

- https://fonts.google.com/
- E.g. to include font **Lato** in Google Fonts:
  - to embed your selected fonts into a webpage, copy this code into the <head> of your HTML document:

    ```
    <link href="https://fonts.googleapis.com/css?family=Lato" rel="stylesheet">
    ```
  - Use the following CSS rules:

    ```
    font-family: 'Lato', sans-serif;
    ```

Exercise



HEADER

GALLERY

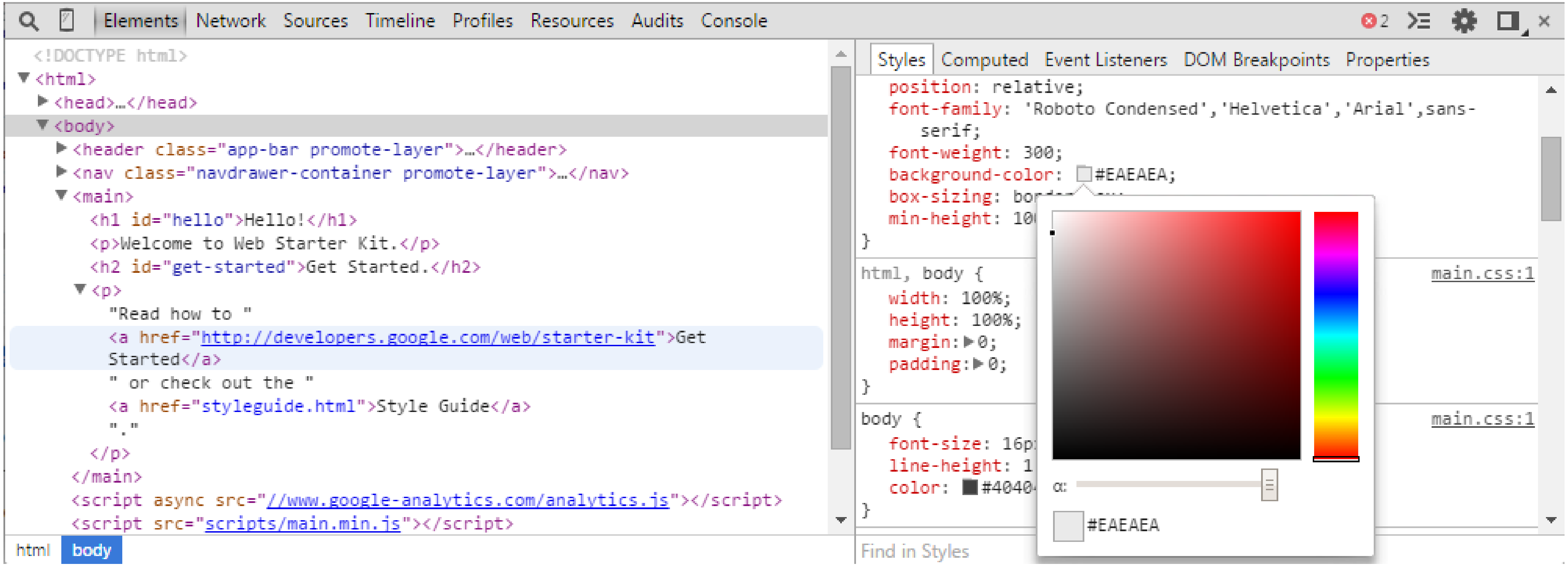
LEFT-SIDEBAR

CONTENT

RIGHT-SIDEBAR

FOOTER

Dev Tools (F12)



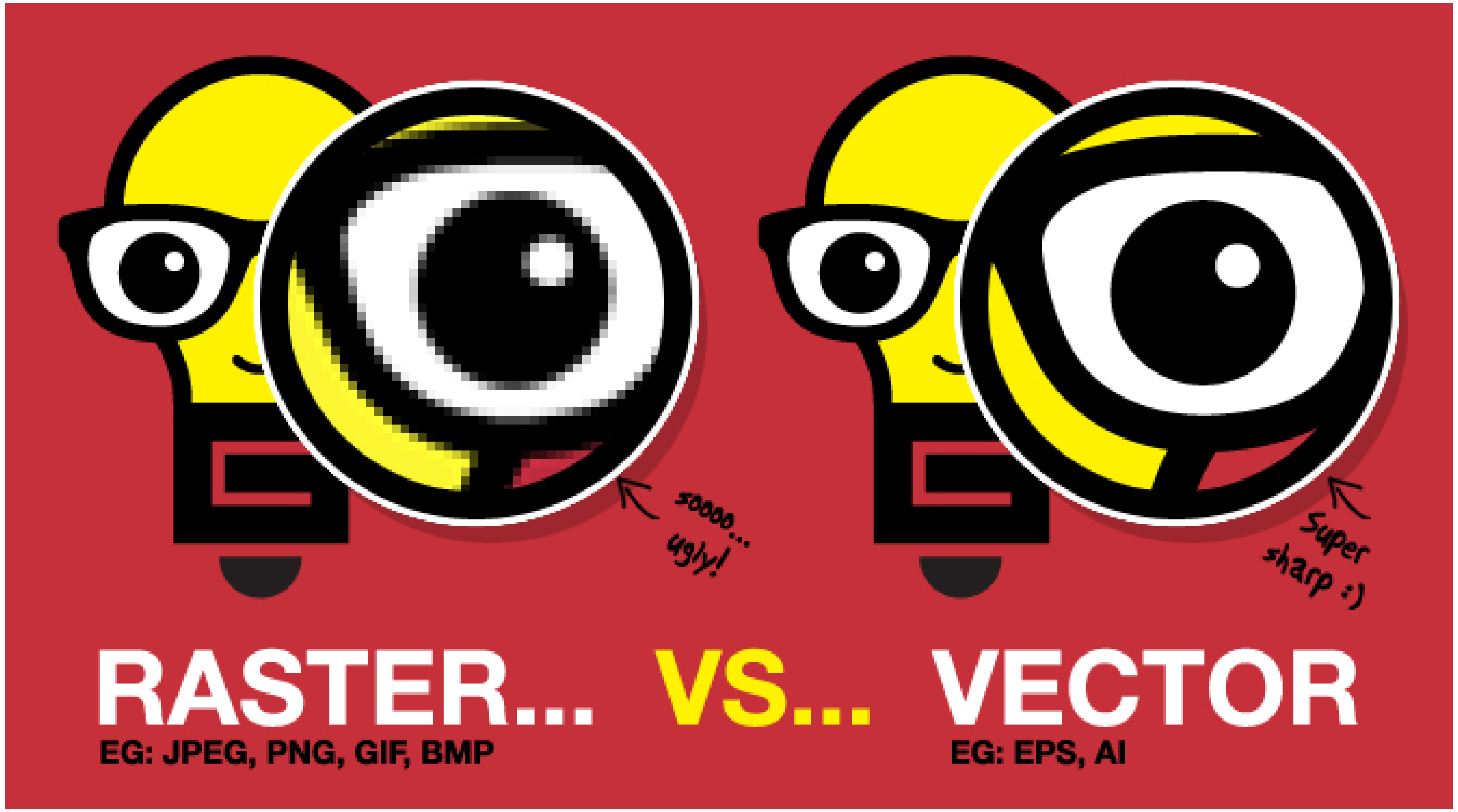
```
<!DOCTYPE html>
▼<html>
  ▶<head>…</head>
  ▼<body>
    ▶<header class="app-bar promote-layer">…</header>
    ▶<nav class="navdrawer-container promote-layer">…</nav>
    ▼<main>
       <h1 id="hello">Hello!</h1>
       <p>Welcome to Web Starter Kit.</p>
       <h2 id="get-started">Get Started.</h2>
      ▼<p>
         "Read how to "
         <a href="http://developers.google.com/web/starter-kit">Get
         Started</a>
         " or check out the "
         <a href="styleguide.html">Style Guide</a>
         "."
       </p>
     </main>
     <script async src="//www.google-analytics.com/analytics.js"></script>
     <script src="scripts/main.min.js"></script>
```

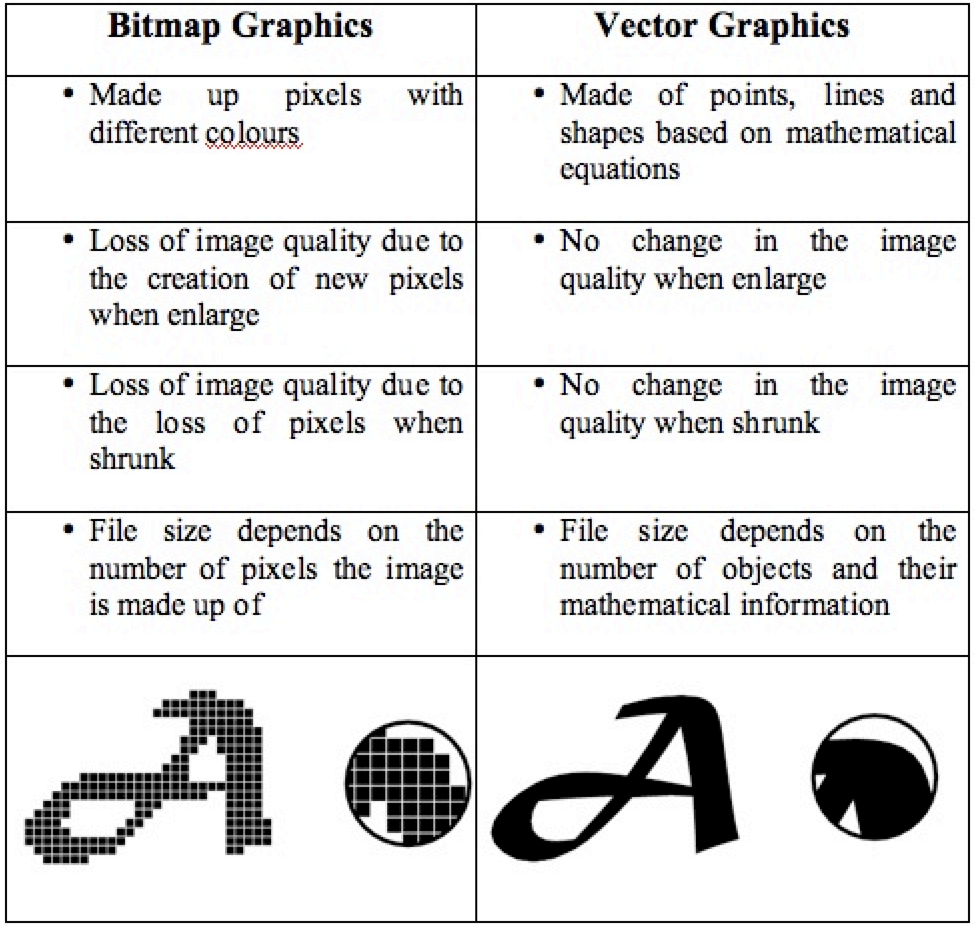
```
  position: relative;
  font-family: 'Roboto Condensed','Helvetica','Arial',sans-
     serif;
  font-weight: 300;
  background-color: □#EAEAEA;
  box-sizing: bor
  min-height: 10
}
html, body {                                   main.css:1
  width: 100%;
  height: 100%;
  margin:▶0;
  padding:▶0;
}
body {                                         main.css:1
  font-size: 16p
  line-height: 1
  color: ■#4040
}
```

□ #EAEAEA

https://developer.chrome.com/devtools

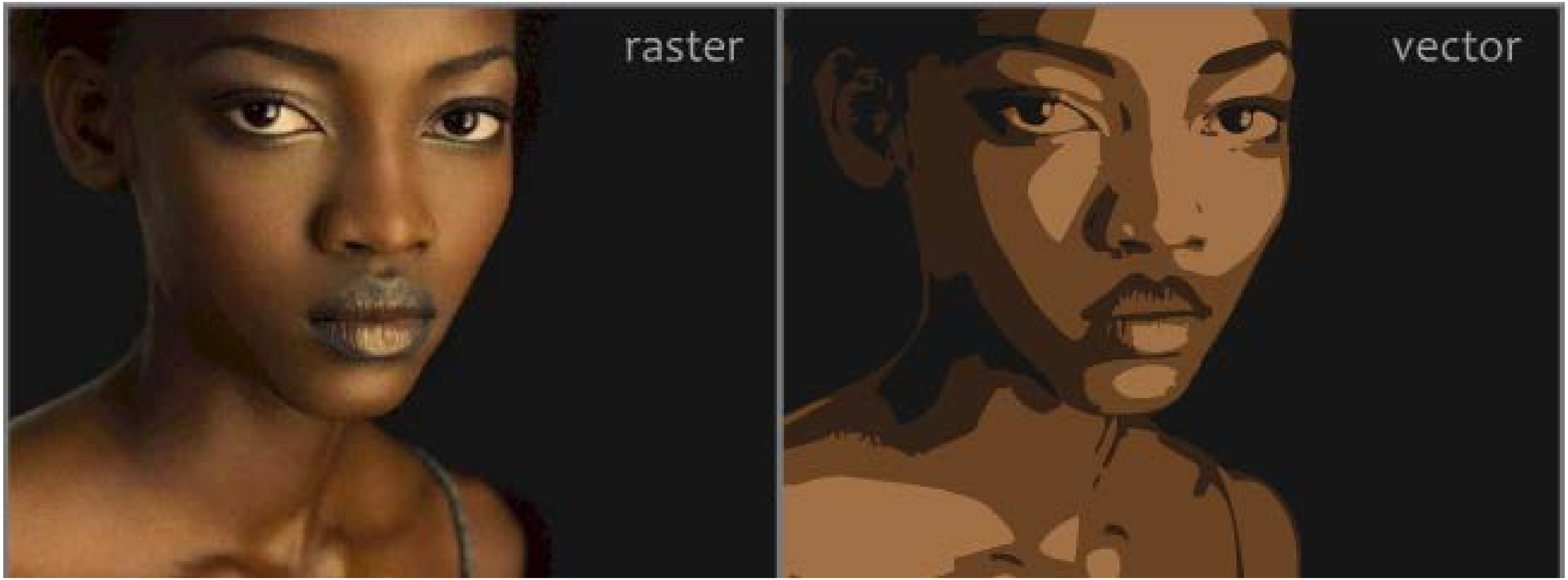
RASTER vs VECTOR





Bitmap Graphics	Vector Graphics
• Made up pixels with different colours	• Made of points, lines and shapes based on mathematical equations
• Loss of image quality due to the creation of new pixels when enlarge	• No change in the image quality when enlarge
• Loss of image quality due to the loss of pixels when shrunk	• No change in the image quality when shrunk
• File size depends on the number of pixels the image is made up of	• File size depends on the number of objects and their mathematical information

THE WIZARD OF VECTOR



http://vectormagic.com/

CSS Positioning Example

http://codepen.io/jamesbarnett/pen/CELir

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

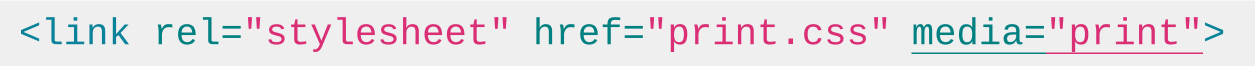
How do you select all p elements inside a div element?

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

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

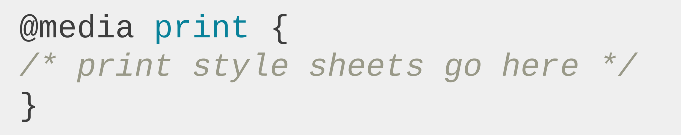
More: media queries for printing

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

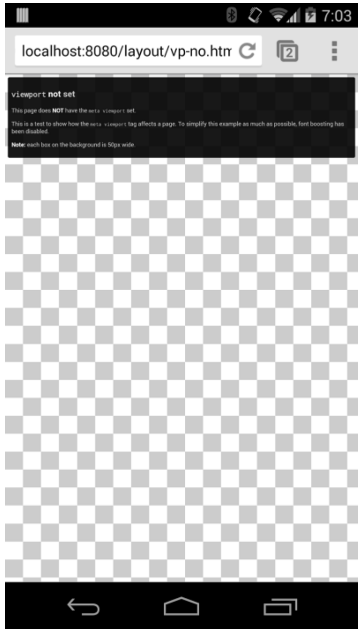
Responsivity with css

1
2
3
1
2
3
4
5
6
7
8
9
10

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">
```



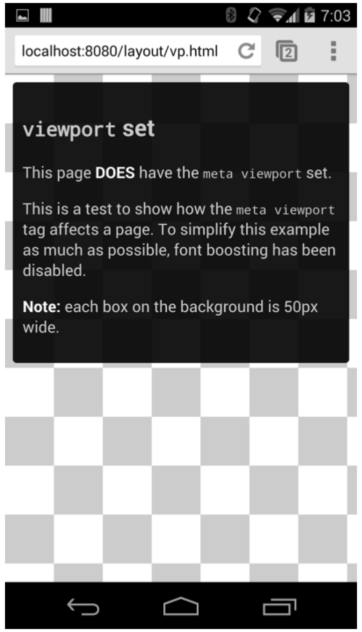
`viewport` **not set**

This page does **NOT** have the `meta viewport` set.

This is a test to show how the `meta viewport` tag affects a page. To simplify this example as much as possible, font boosting has been disabled.

Note: each box on the background is 50px wide.

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.



`viewport` **set**

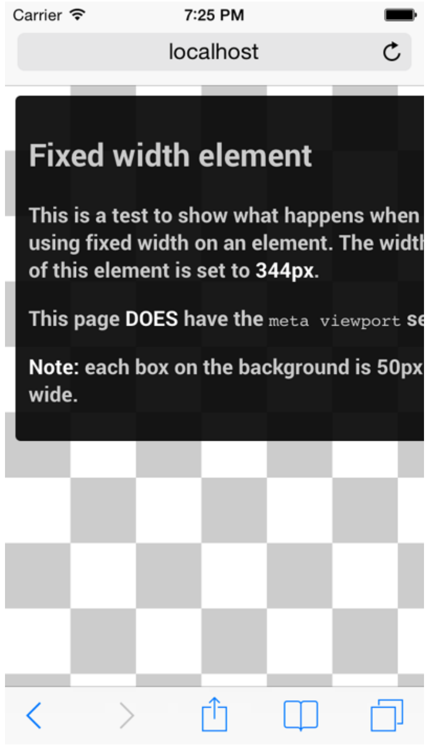
This page **DOES** have the `meta viewport` set.

This is a test to show how the `meta viewport` tag affects a page. To simplify this example as much as possible, font boosting has been disabled.

Note: each box on the background is 50px wide.

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.



Fixed width element

This is a test to show what happens when using fixed width on an element. The widt of this element is set to **344px**.

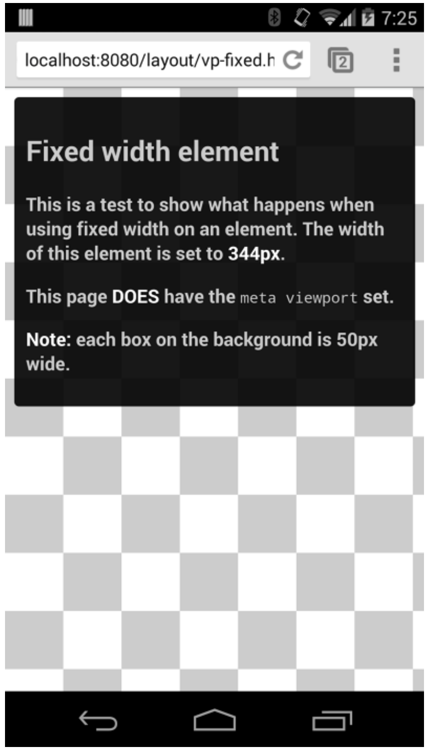
This page DOES have the `meta viewport` se

Note: each box on the background is 50px wide.

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.



Fixed width element

This is a test to show what happens when using fixed width on an element. The width of this element is set to **344px**.

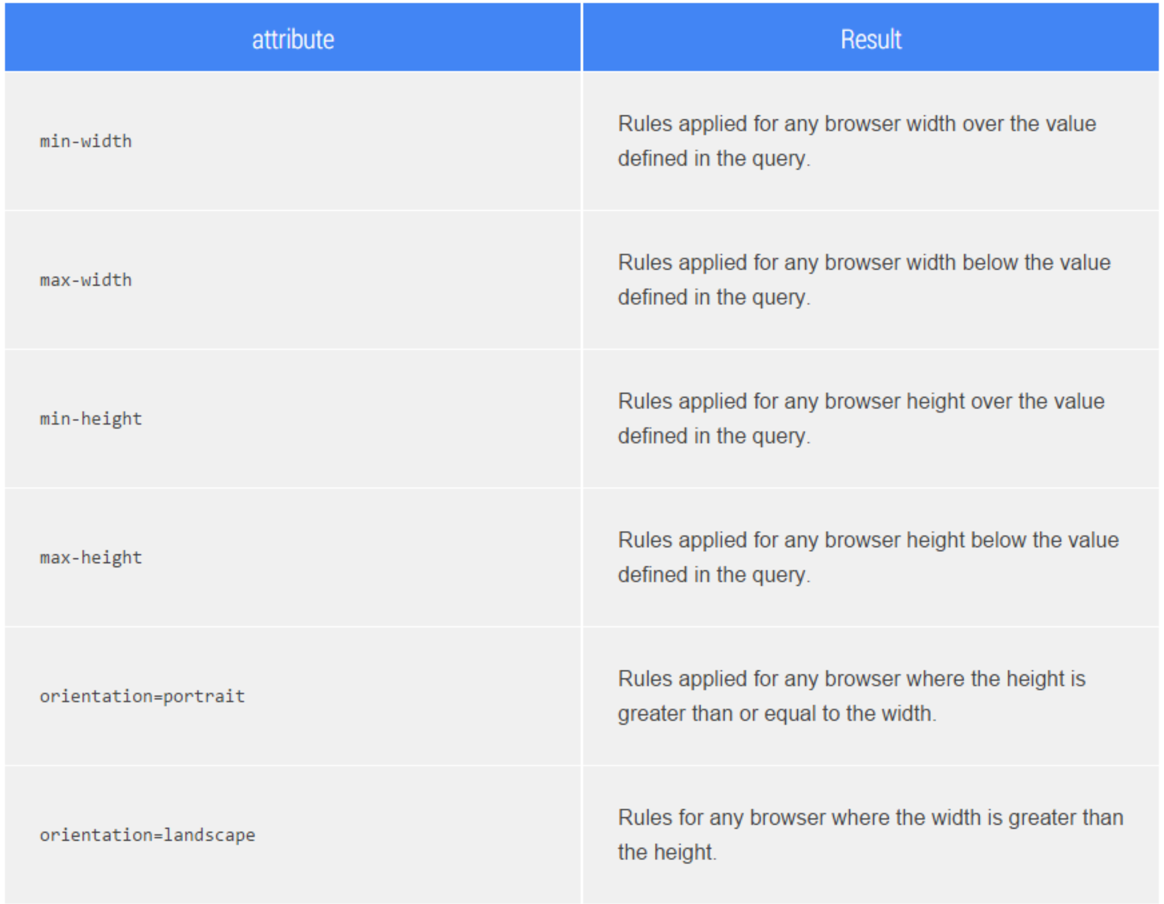
This page DOES have the `meta viewport` set.

Note: each box on the background is 50px wide.

3) Use media queries

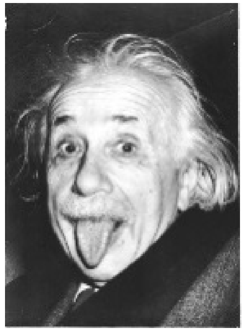
Media queries enable us to create a responsive experience, where specific styles are applied to small screens, large screens and anywhere in between. The media query syntax allows for the creation of rules that can be applied depending on device characteristics.

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

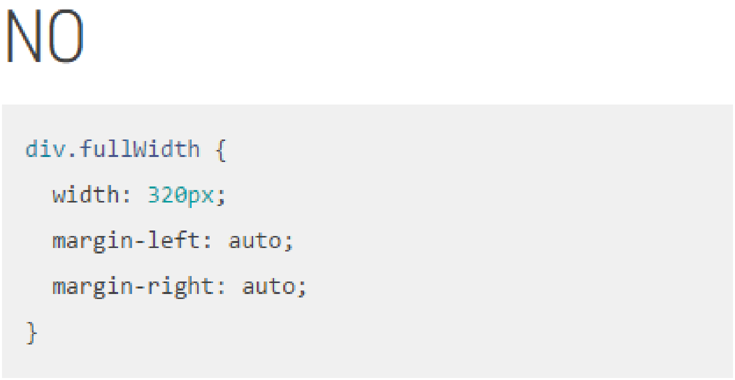


attribute	Result
`min-width`	Rules applied for any browser width over the value defined in the query.
`max-width`	Rules applied for any browser width below the value defined in the query.
`min-height`	Rules applied for any browser height over the value defined in the query.
`max-height`	Rules applied 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!

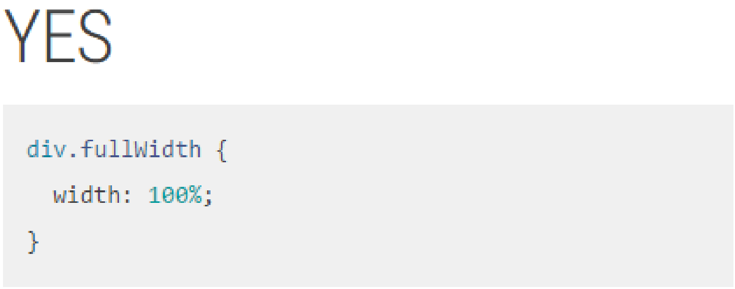


- A key concept behind responsive design is fluidity and proportionality as opposed to fixed width layouts. Using relative units for measurements can help simplify layouts and prevent accidentally creating components that are too big for the viewport.
- For example, setting width: 100% on a top level 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 */ }
```

On the web – in case you missed this class

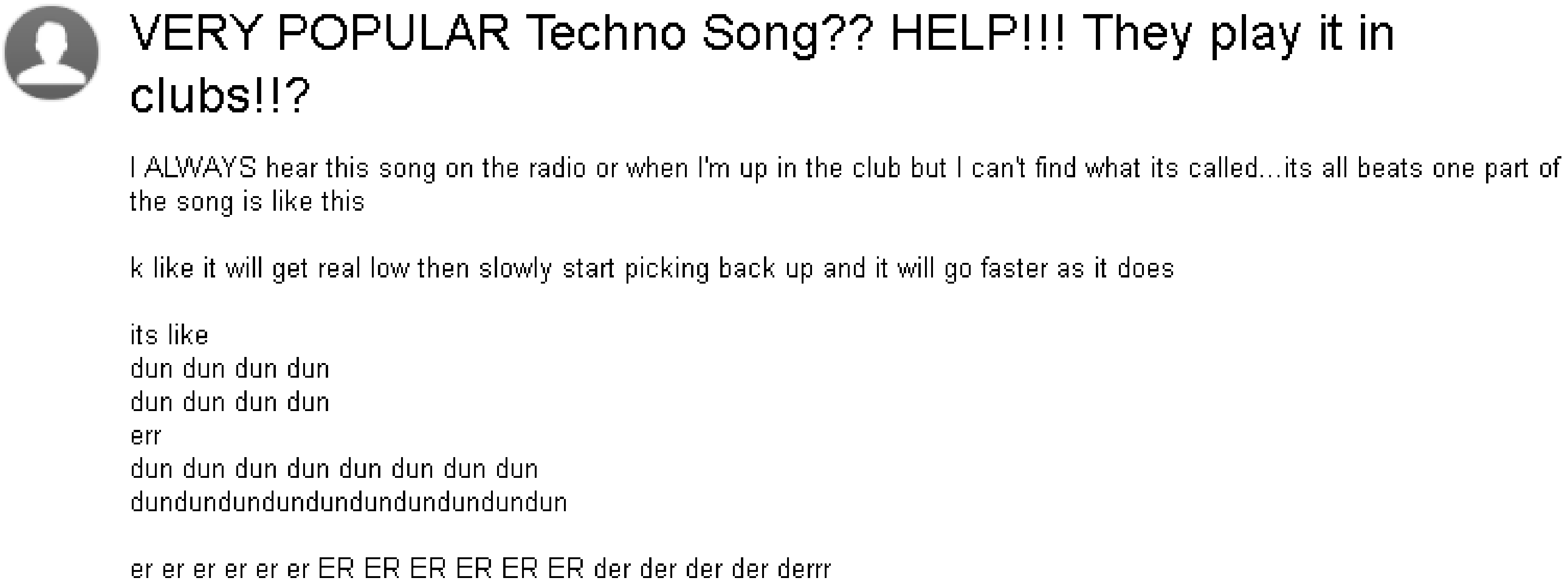
- https://developers.google.com/web/fundamentals/ (en - free)
- http://www.codecademy.com/tracks/web (en - free)
- http://www.html.it/guide/guida-html5/ (ita - free)
- https://www.codeschool.com/paths/html-css (en)
- http://teamtreehouse.com/tracks/front-end-web-development (en)

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

CSS
IS
AWESOME

Being stuck?

0. Google (Stack Overflow suggested, Yahoo Answers not 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



VERY POPULAR Techno Song?? HELP!!! They play it in clubs!!?

I ALWAYS hear this song on the radio or when I'm up in the club but I can't find what its called...its all beats one part of the song is like this

k like it will get real low then slowly start picking back up and it will go faster as it does

its like
dun dun dun dun
dun dun dun dun
err
dun dun dun dun dun dun dun dun
dundundundundundundundundundun

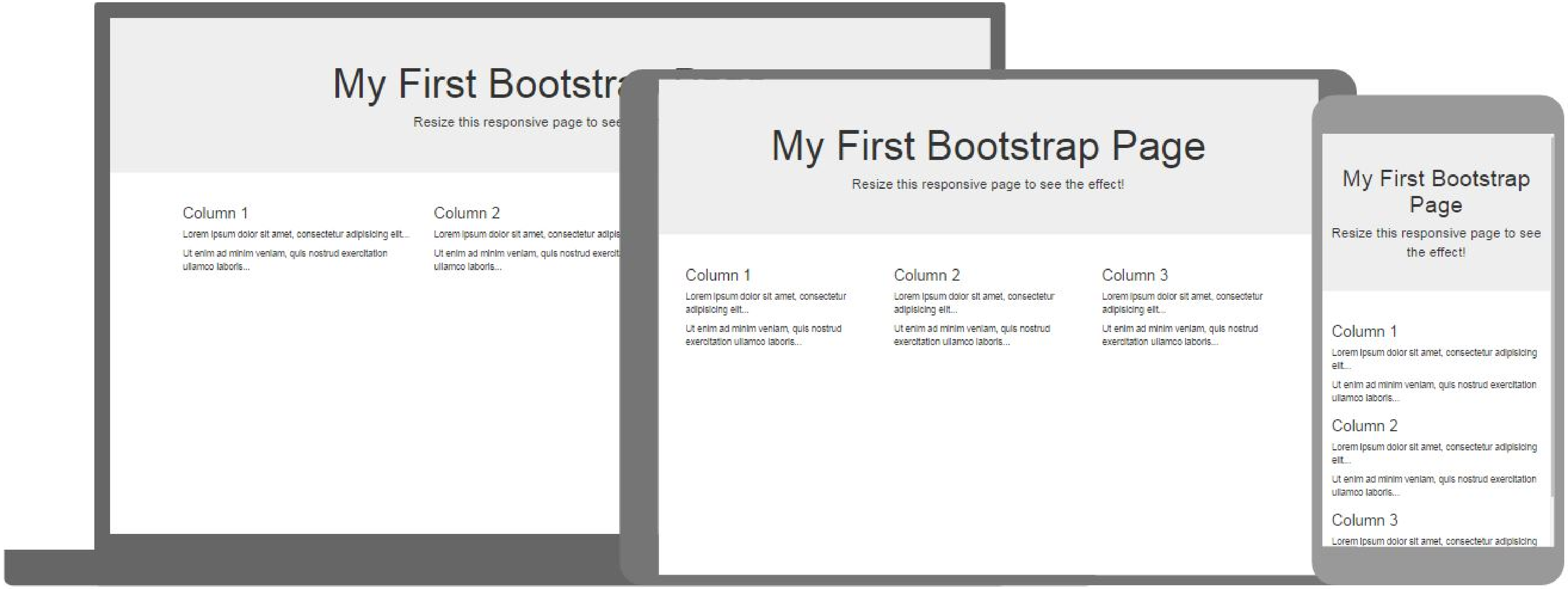
er er er er er er ER ER ER ER ER ER der der der der derrr

Bootstrap

- **Bootstrap** is an HTML5 & CSS3 framework designed to help you kickstart the development of webapps 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



My First Bootstrap Page

Resize this responsive page to see the effect!

Column 1

Lorem ipsum dolor sit amet, consectetur adipisicing elit...

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris...

Column 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit...

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris...

Column 3

Lorem ipsum dolor sit amet, consectetur adipisicing elit...

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris...

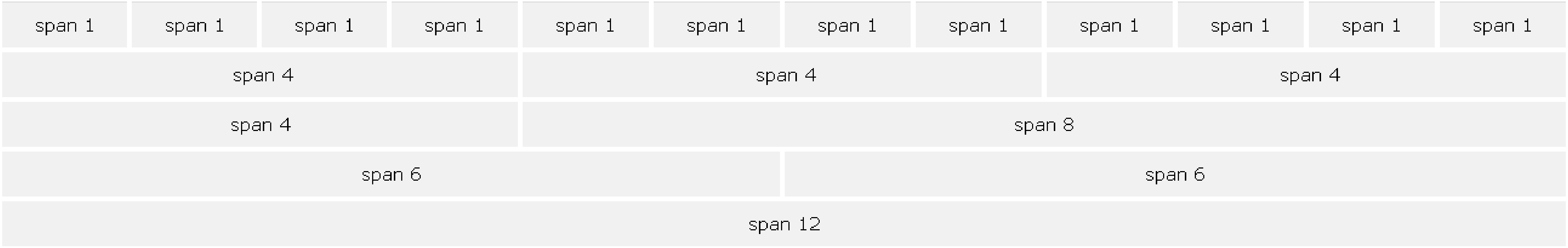
Why Use Bootstrap? Advantages of Bootstrap:

- **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/bootstrap.min.css">

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

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

Bootstrap Grid System



<table>
<tr><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td></tr>
<tr><td colspan="4">span 4</td><td colspan="4">span 4</td><td colspan="4">span 4</td></tr>
<tr><td colspan="4">span 4</td><td colspan="8">span 8</td></tr>
<tr><td colspan="6">span 6</td><td colspan="6">span 6</td></tr>
<tr><td colspan="12">span 12</td></tr>
</table>

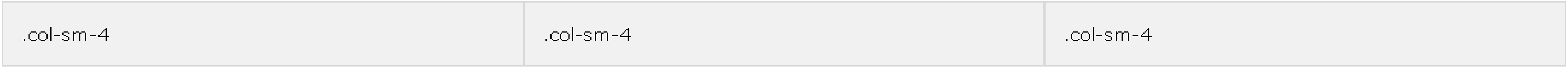
The Bootstrap grid system has four classes:

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

Basic Structure of a Bootstrap Grid

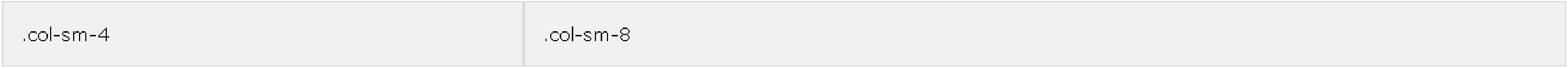
```
<div class="row">
  <div class="col-*-*"></div>
</div>
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>
<div class="row">

</div>
```

Example: Equal and Unequal Columns



.col-sm-4 .col-sm-4 .col-sm-4

```
<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>
```



.col-sm-4 .col-sm-8

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

Tutorial

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

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