Introduction to Web Development

Cognitir

Cognitir was founded in August 2015 by David Haber and Neal Kumar. The company provides data science and web development courses designed for business and non-technical professionals. Given David's technical expertise and Neal's vast business & training experience both founders have combined their backgrounds to create world-class, practical training courses.

We always strive to create best-in-class products, so feel free to email us any comments on this course and/or new product ideas. Our email is **feedback@cognitir.com**.

We sincerely hope that you enjoy taking the course as much as we enjoyed creating it!

 *@cognitir*

 *Cognitir*

 *Cognitir*

Terms and Conditions For This Course

These notes are proprietary to Cognitir.

Only the intended receiver of the course ticket ("attendee"), as confirmed on Cognitir LLC's attendance list, has the right to print these electronic course notes once. The attendee may hold and access these electronic course notes indefinitely as long as he/she abides by the Terms and Conditions listed on this page and the Cognitir LLC website. The Terms and Conditions on Cognitir's website can be found at:

http://www.cognitir.com/terms

The Terms and Conditions on this page and the website will remain in full effect before, during, and after the course.

Beyond printing these course notes once, the attendee is prohibited from distributing, copying, sharing, duplicating, and/or altering this file in any way without the expressed written consent of Cognitir.

Cognitir reserves the right to take full legal action against attendees and affiliates of attendees who do not abide by Cognitir's Terms and Conditions listed on this page and the website.

By enrolling in and attending this course, the attendee agrees to fully abide by the Terms and Conditions listed on this page and Cognitir's website.

Authors: David Haber

Version: 1.2

For more information: www.cognitir.com

This Course

Designed for non-technical and business professionals:

Cognitir provides a balanced framework of technical skills and business problem solving to enhance decision making and product development.

Course Topics:

- How the internet works
- Build a fully responsive, interactive website using HTML, CSS, and JavaScript
- Integration of third-party tools and website analytics

Course Goal:

Learn the basics of web development to build your own product/project landing pages.

Course Outline

Morning Session (09:00 – 12:30)

- Introduction
- The Internet
- Basics of HTML: Creating Our First Website
- Introduction to CSS

Lunch (12:30 – 13:30)

Afternoon Session (13:30 – 16:45)

- Responsiveness: Learn how to build a website that is multi-device ready
- JavaScript: Adding Interactivity to Our Website
- Making Your Landing Page Available Online
- Techniques on how to measure the success of your landing page

Wrap-up (16:45 – 17:00)

- Q&A

Why HTML, CSS, and JavaScript?

- Services such as **WordPress** offer easy-to-use templates for website creation.
- **BUT:** Understanding the **internet** and **basic web technologies** such as HTML, CSS, and JavaScript means that you can
  - **customize** your own website to perfectly fit your needs,
  - **evolve** your website as your project/business grows, AND
  - **reduce** costs that you would otherwise spend on fancy WordPress templates, etc.
- *Therefore, it is completely worth it to learn how to create webpages using HTML, CSS, and JavaScript*.

Let's Start Hacking!

CNN Tech » Gadgets | Cyber Security | Innovation Nation | Vital Signs

Space + science

NASA re-establishes contact with missing vessel

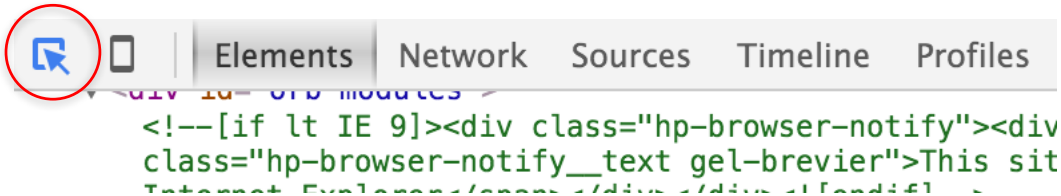
Students build awesome websites!

Innovate

Let's Start Hacking! (cont'd)

Exercises:

1. Open **Google Chrome** on your computer.
2. Go to **www.cnn.com**.
3. **Right-click** anywhere on the page and select "**Inspect element**" to open the developer tools.
4. Select the **left-most item** from the options as shown in the following screenshot:



5. Hover over different elements on the web page.
6. Modify some of the content in the developer tools window.

"If you can change technology, you can change the world."

Susan Wojcicki, CEO of YouTube

THE INTERNET

OVERVIEW

- What is the difference between a computer network, the internet, and World Wide Web?
- What is an Internet Service Provider?
- What actually happens when you visit a website?
- How is a server and client different?
- What is HTTP?
- Which role do DNS, IP addresses and URLs play?
- What is “Coding”?

Computer Network

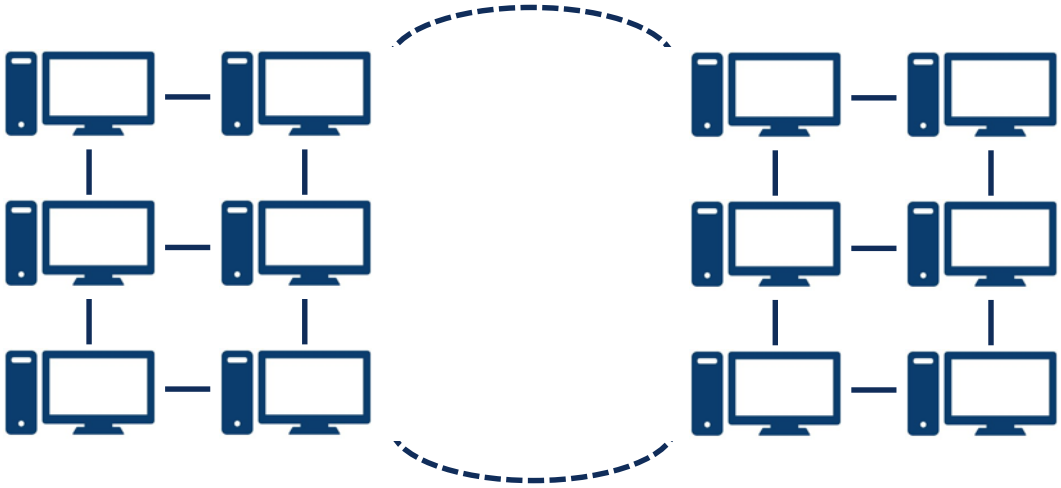
If we connect a computer to another computer, we create a **computer network.**

Computers can be connected via **cables** or **wirelessly**.

Computers communicate via **network packets**.

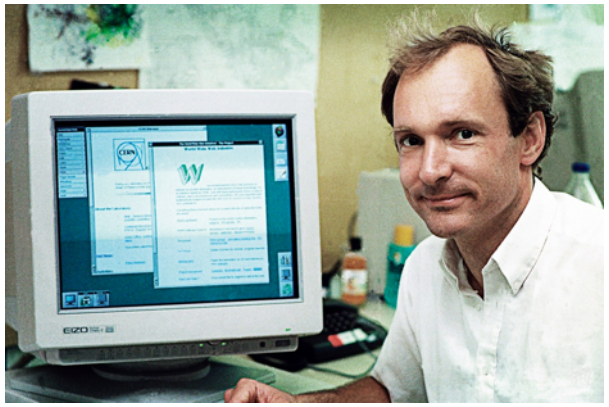
The Internet

- The internet is a **networks of networks**.
- It connects millions of computer together so that they can **share information across the world**.

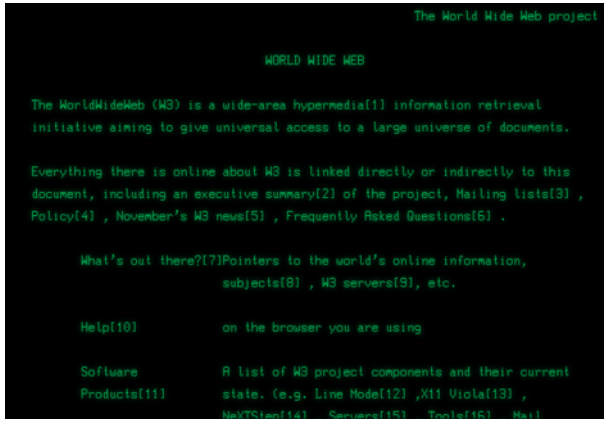


World Wide Web

- The **World Wide Web** is essentially just another way through which **information can be shared over the internet**.
- It's a **collection of pages of information** on computers that are linked together across the world.
- **The Web** was invented by **Tim Berners-Lee** in 1989. He wrote the first **Web browser** at CERN in Switzerland in 1990.



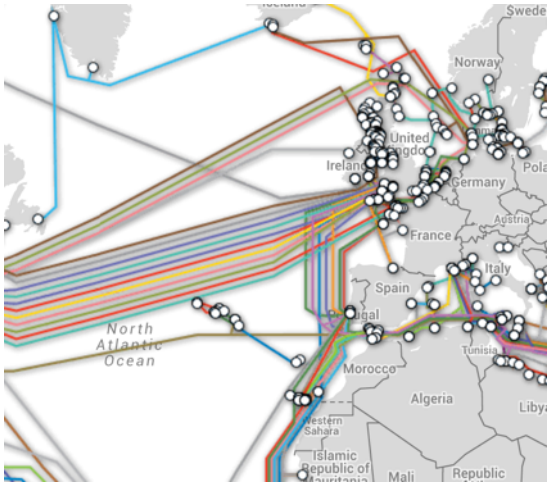
Tim Berners-Lee



http://line-mode.cern.ch/www/hypertext/WWW/TheProject.html

How Are You Connected?

- Your computer connects to the internet via your **internet service provider** (ISP).
- You can **access the web with a web browser**, which is a software application for **retrieving**, **presenting,** and **traversing** information resources on the World Wide Web.

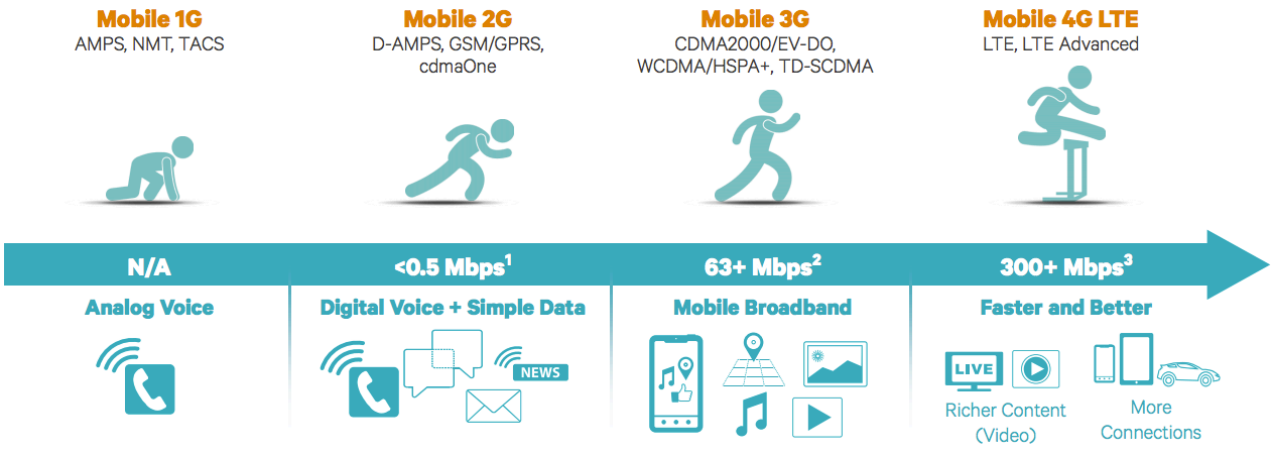


submarinecablemap.com

Cableship “Asean Restorer”

Remember this?
https://en.wikipedia.org/wiki/File:Dial_up_modem_noises.ogg

Mobile Internet



Source: Qualcomm – The Evolution of Mobile Technologies

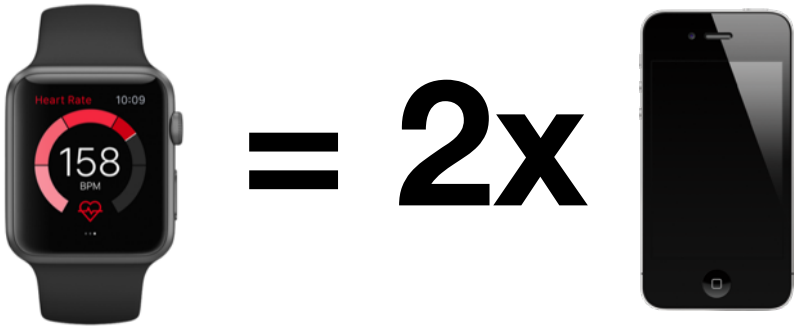
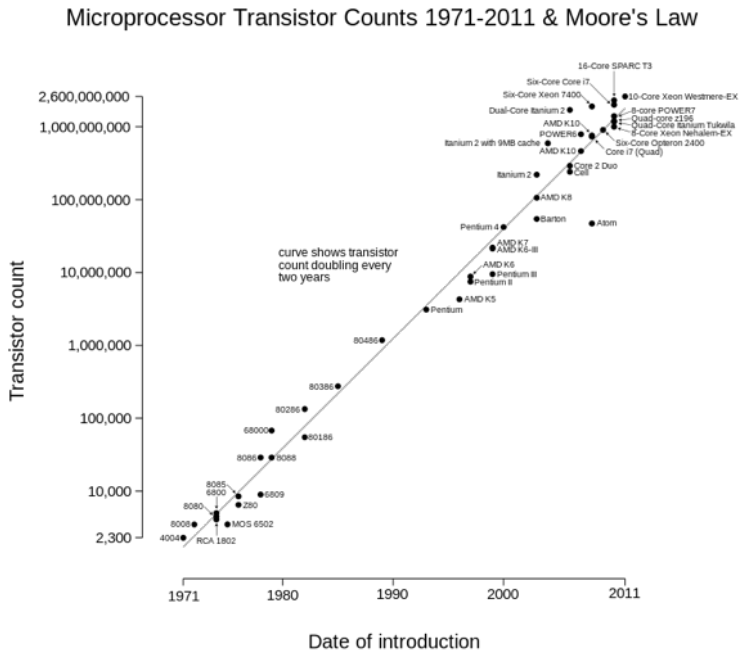
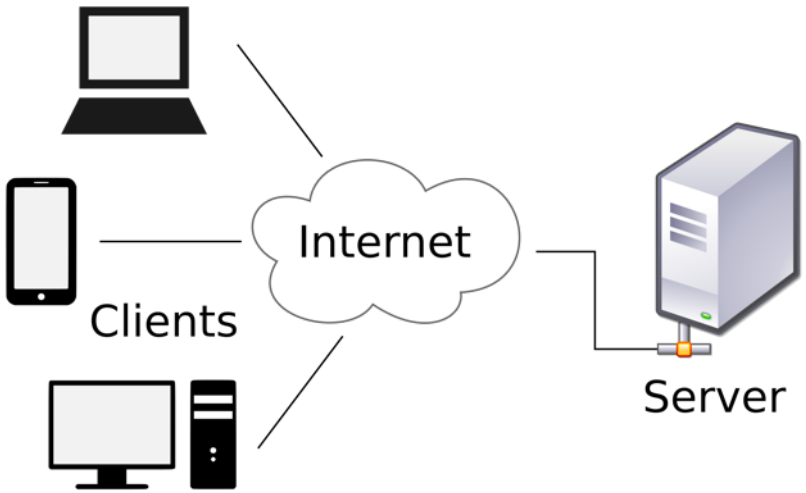


Image source: Apple



Client & Server

- A **client** is the computer that **requests** information.
- A **server** is a computer in the network that **listens** to incoming requests and **provides** the information requested.



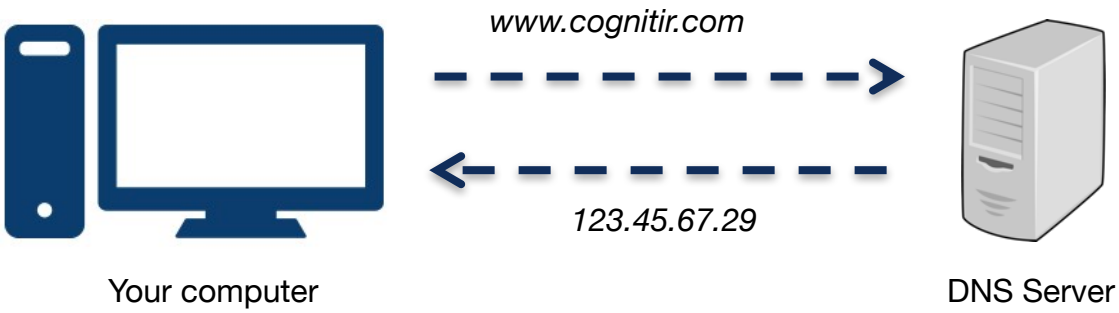
What is HTTP?

- Computers need a **protocol** to exchange information. Think of protocol as an **agreement** as to how the information will be structured and how each side will send and receive it.
- **Hyper Text Transfer Protocol (HTTP)** is a type of protocol.
- Clients send **HTTP requests**, servers send **HTTP responses**.
- Other protocols for the internet include **https**, **ftp**, **imap**, **pop**, and **smtp**.

HTTP is the language that computers use to communicate over the internet.

DNS, IP Addresses, and URLs

- Each computer connected to the Internet must have a **unique address**. This address is known as an **IP address**.
- IP addresses are not easy to remember for humans. We use **URLs (Uniform Resource Locators)** to access resources on the Web.
- **DNS (Domain Name System)** translates **URLs into IP addresses**. The DNS system is, in fact, its own network. If one DNS server does not know how to translate a particular domain name, it asks another one, and so on, until the correct IP address is returned.



IP addresses are in the form nnn.nnn.nnn.nnn where nnn must be a number from 0 - 255.

What actually happens when you type in www.google.com?

1. Browser checks if web resource is already stored on computer (cached). If it is, skip to 7.
2. Browser asks operating system (OS) for IP address that corresponds to *www.google.com*.
3. Operating system looks up corresponding IP address and returns IP address to browser.
4. Browser opens a connection to *Google* and sends request.
5. Browser receives response from *Google*.
6. Browser checks response.
7. Browser decides action based on response (Is it a web page? Is it an image?).
8. Browser displays response on screen.

What is Coding?

- A **computer program** is a list of instructions that tell the computer what to do.

- **Coding** or **programming** is the art of creating these instructions.

- We write our computer programs in specific **programming languages**.

Programming Language Trends on Github:
https://github.com/blog/2047-language-trends-on-github

SUMMARY

- What is the difference between a computer network, the internet, and World Wide Web? ✓
- What is an Internet Service Provider? ✓
- What actually happens when you visit a website? ✓
- How is a server and client different? ✓
- What is HTTP? ✓
- Which role do DNS, IP addresses and URLs play? ✓
- What is “Coding”? ✓

MORE INFO

- Introduction to DNS
  https://technet.microsoft.com/en-us/library/cc958978.aspx

“A computer once beat me at chess, but it was no match for me at kick boxing."

Emo Philips, Comedian

INTRODUCTION TO HTML

OVERVIEW

- What is HTML and what do we use it for?
- How do we use HTML to create our first web page?
- What are HTML tags?
- How can we add text, images and hyperlinks to our web pages?
- What is the difference between absolute and relative hyperlinks?
- How can we create lists with HTML?
- What is the “Document Object Model”?

What is HTML?

- **HyperText Markup Language** (HTML) is a series of codes typed into a text file to create a website.
- With HTML we can define the **content** and **structure** of a web page.
- We can **define content** including

text	**images**
headlines	**lists**
links	**and much more…**

HTML Basics

- Your **web browser** (e.g., Chrome) knows how to “read” HTML.
- HTML documents use **tags** to tell the browser about structure and content:

- There are many different HTML tags!
- Most of them follow the **same format**.

Exercises:
HTML code is “public”, which means that you can view the HTML page source of any website through your browser.

1. Open **Google Chrome**.
2. Go to your **favorite website** and **right-click** anywhere on the page.
3. Select “**View Page Source**” from the dropdown menu.
4. Make a list of HTML tags that you find in the HTML source.

HTML Headings

- The **<h1>** tag is used to define an HTML **heading**:

```
<h1>I am an important heading</h1>
```

- There are **six "levels"** of headings: <h1> to <h6>

Exercises:

1. Open **Sublime Text Editor**.
2. Add an **<h1>** heading to the document.
3. Save the document as "**index.html**" in a folder on your Desktop.
4. Double-click on the newly created file and wait for your web browser to open. Check if your heading is there.
5. Add a few other headings (e.g., <h2>, <h3>) to your website and see how they differ.

HTML Paragraphs

- The **<p>** tag defines a **paragraph** of text.

```
<p>I am a paragraph</p>
```

- Your browser will automatically add some space (margin) before and after the <p> element.

Exercises:

1. Add some text as a paragraph to your HTML document. You can insert it below the headings you created in the previous exercise.
2. Open **index.html** in Chrome to check your results.
3. Find out what the following tags do:
   a. ****
   b. **<small></small>**
   c. ****

Hyperlinks

- The **<a>** tag defines a **hyperlink** which links web pages together.

```
<a href="http://www.google.com">I link to Google</a>
```

- We can make the hyperlink open the referenced web page in a **new tab**:

```
<a href="http://www.google.com" target="_blank">
    I will open Google in a new tab.
</a>
```

Exercise:

1. Add a **hyperlink** to your HTML document.

HTML Images

- The **<img> tag** defines an image in an HTML document.

```
<img src="http://www.cognitir.com/static/img/classroom.jpg">
```

- Note that it doesn't have a closing tag!
- Only tags with content "inside" them have to be closed (e.g., headlines, paragraphs, tables).

Exercises:

1. Add an **image element** to your HTML document.
2. You can **right-click** on almost any image on the internet and copy its URL. Find an image on the internet, copy its URL, and embed it on your web page.

HTML Attributes

- HTML tags can have **attributes** that provide **additional information** about the tag.

- Examples:
  - Where should a hyperlink take me?
  - Which image should be displayed?

```
<a href="http://www.google.com">I link to Google</a>

<img src="http://www.cognitir.com/static/img/classroom.jpg">
```

- Attributes are always **specified in the start tag**.
- Attribute names are specified as **name = “value” pairs**.

Relative vs. Absolute URLs

- There are two different types of URLs: **absolute** and **relative**.
- A **absolute** URL can be used anywhere on the internet, not just on a single web page, e.g.,

```
<a href="http://www.google.com">I link to Google</a>

<img src="http://www.cognitir.com/static/img/classroom.jpg">
```

- A **relative** URL specifies the name of a file in the context of the current document, e.g.,

```
<a href="about.html”>Link to the about page in same folder</a>

<img src="image.png">
```

- **Unordered** and **ordered lists** are often used in HTML.

An Unordered List

```
<ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ul>
```

We use one tag to define the beginning and end of the list.

An Ordered List

```
<ol>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ol>
```

We use one tag per element in the list.

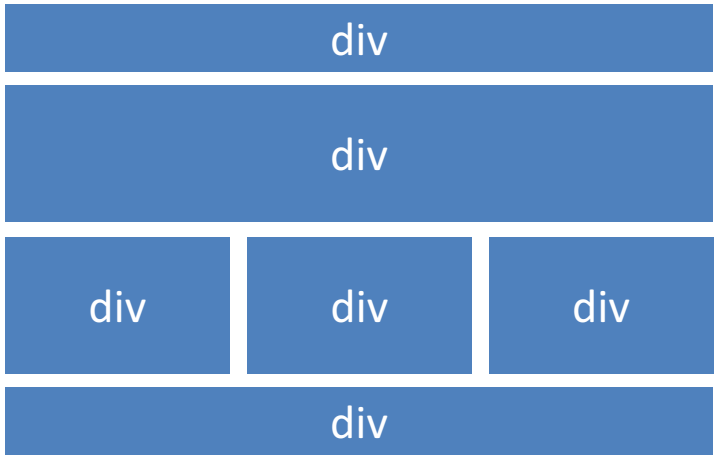
Exercises:

1. Add an **unordered list** to your HTML document.
2. Add the following attribute to the starting tag of your list and observe its behavior: ***style="list-style-type:square"***

HTML Containers

** ** elements define a **section** in an HTML document.

It is often used to **group elements**, **apply CSS styling, manipulate with JavaScript** (more on this later!).




```
<div>
    <h4>Product Features</h4>
    <p>Our product has state-of-the-art features.</p>
</div>
```

Exercises:

1. Add a ** ** element to your HTML document.
2. If you **add another ** element after the one you added in 1., what can you say about its positioning?

HTML Containers (cont'd)

- **** is similar to ** ** but often used to group smaller parts of HTML together.
- It does not add a new line after element. We say it's **in-line**.

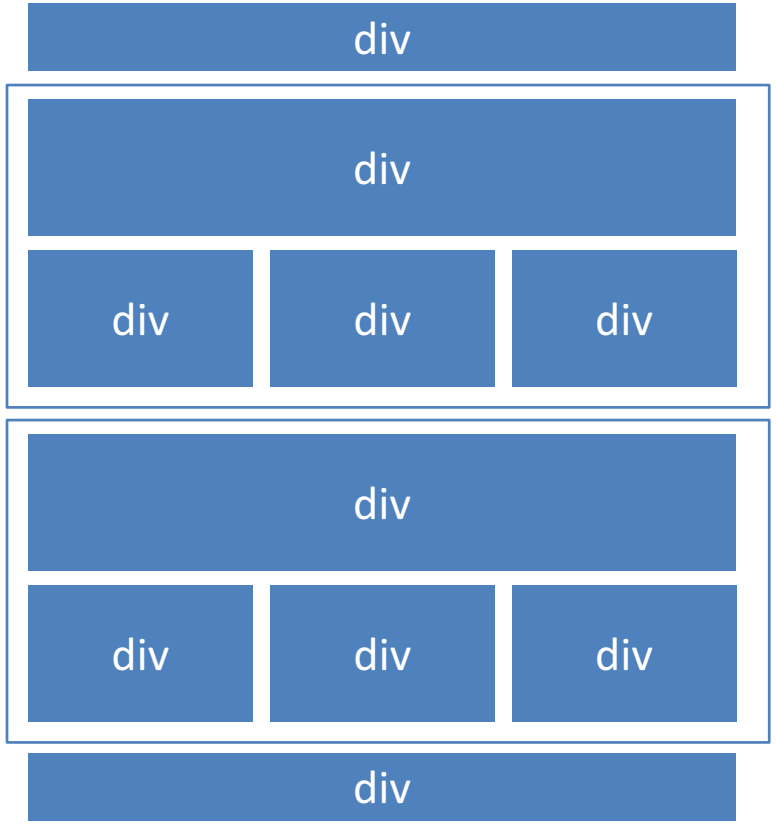
```
<div>
    <h4>Product Features</h4>
    <p>
        Our product has <span style="color:red">state-of-the-
        art</span> features.
    </p>
</div>
```

Exercise:

1. Add the **** element shown above to your HTML document.
2. **Refresh** the page.
3. What can you say about the positioning of the element?

HTML Containers (cont'd)

- The **<section>** tag is used for thematic grouping of content, typically with a heading.
- A web page could be **split into sections**, e.g.,:
  - Introduction
  - Product Features
  - Team
  - Contact Information
- A section often contains multiple elements.
- Other common container elements are:
  - <header>
  - <footer>
  - <article>



HTML Structure

- HTML documents generally follow this **structure**:

```
<html>
    <head>
        …
    </head>
    <body>
        …
    </body>
</html>
```

<html></html> marks the start and end of the HTML document.

We place additional information (metadata) about our document in the <head> element.

We place our content in the body section (this is what we want to display on screen)

- The following tags describe **metadata**:
  <title>, <style>, <meta>, <link>, <script>, and <base>

Exercises:

1. Change your HTML document so that it follows the structure shown above. Everything that you have added to your HTML document so far should be put **in the <body> element**.
2. Reload the page to make sure that it still displays correctly.

Doctype Declaration

- **<!DOCTYPE html>** tells the browser **which version of HTML you are using**:

```
<!DOCTYPE html>
<html>
    ...
</html>
```

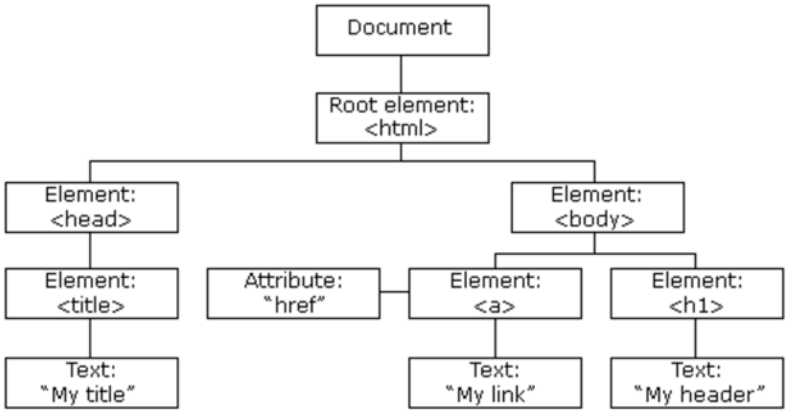
- This is the declaration for **HTML5**, the most recent version.
- If you don't put the doctype, some (especially older) browsers may not display your content properly!

Exercise:

1. Add the **doctype declaration** as the first line to **index.html**.

Document Object Model

- The **HTML DOM** is a tree-based diagram that lays out the elements of an HTML document.



- The DOM is a **W3C (World Wide Web Consortium) standard**.
- The DOM defines a **standard for accessing documents**.

SUMMARY

- What is HTML and what do we use it for? ✓
- How do we use HTML to create our first web page? ✓
- What are HTML tags? ✓
- How can we add text, images and hyperlinks to our web pages? ✓
- What is the difference between absolute and relative hyperlinks? ✓
- How can we create lists with HTML? ✓
- What is the “Document Object Model”? ✓

MORE INFO

- HTML Tag Reference
  http://www.w3schools.com/tags/
- Absolute vs. Relative Paths/Links
  http://www.coffeecup.com/help/articles/absolute-vs-relative-pathslinks/

“The new information technology…Internet and e-mail…have practically eliminated the physical costs of communication."

Peter Drucker, Management Consultant, Author

INTRODUCTION TO CSS

OVERVIEW

- What is CSS and why is it useful?
- How do CSS and HTML work together?
- What is a CSS rule?
- Which CSS rule can we apply?
- Fonts & Colors
- What is a CSS selector and how do we use it for styling?

- **Cascading StyleSheets (CSS)** allow us to control the **style & layout** of our web pages.

- We is used CSS to add **styling**:
  - Fonts
  - Colors
  - Background images
  - Borders
  - Gradients

BBC website without CSS

- And rich **layouting**:
  - Positioning
  - Alignment
  - Margins

BBC website with CSS

How to Use CSS?

CSS can be added to the **<head> section** of your HTML document.

```
<html>
    <head>
        <style>
            h1 {
                color: blue;
            }
        </style>
    </head>
    <body>
        …
    </body>
</html>
```

The <style> tag is used add CSS definitions to an HTML document.

An example of a CSS definition. Adding this to your document colors all <h1> elements in blue.

Exercises:

1. Add **CSS code** to your HTML document such that it colors all ***<h1> elements in blue***.
2. Add **another CSS rule** such that all your **<p> tags are in red**.

A **CSS rule** consists of a **selector** and a **declaration block**:

Selector
The selector chooses the HTML element which you want to style.

Declaration Block
This is where we define the style. The declaration block contains declarations separated by a semicolon.

Declaration
Each declaration includes a property and a value.

Readability

- Coding is all about **collaboration**!
- It is important that **other people can read your code**.
- Make sure your code is **correctly indented** to enhance readability.
- You can add **comments** to your CSS code:

```
p {
    /* A comment that explains something */
    color: green;
    font-size: 15px;
}
```

Exercise:

1. Make sure that your HTML code in index.html is **correctly indented**.

Where to Put CSS Code?

Header Style Definition: *see slide 44*.

Inline Style Definition:

```
<p>
    I am building my <span
    style="color:red">first</span>
    website.
</p>
```

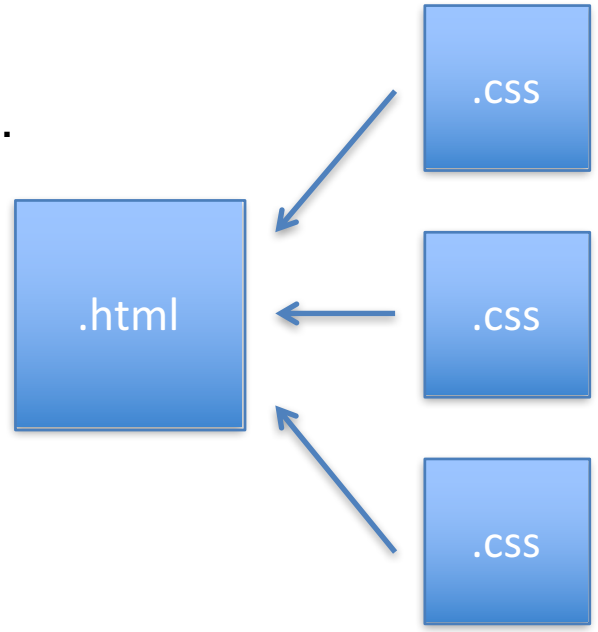
I am building my first website.

External Style Definition:

Create style.css which contains CSS rules.

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

style.css must **only** contain CSS rules!



Changing background color:

```
p {
    background-color: yellow;
}
```

Changing font size:

```
p {
    font-size: 30px;
}
```

Exercises:

1. Add the **above rules** to your HTML document. Note that you can put both CSS declarations (background-color, font-size) into one p selector.
2. How can we **change the background color** of the entire web page?

Adding **space around elements**:

```
img {
    margin: 20px;
}
```

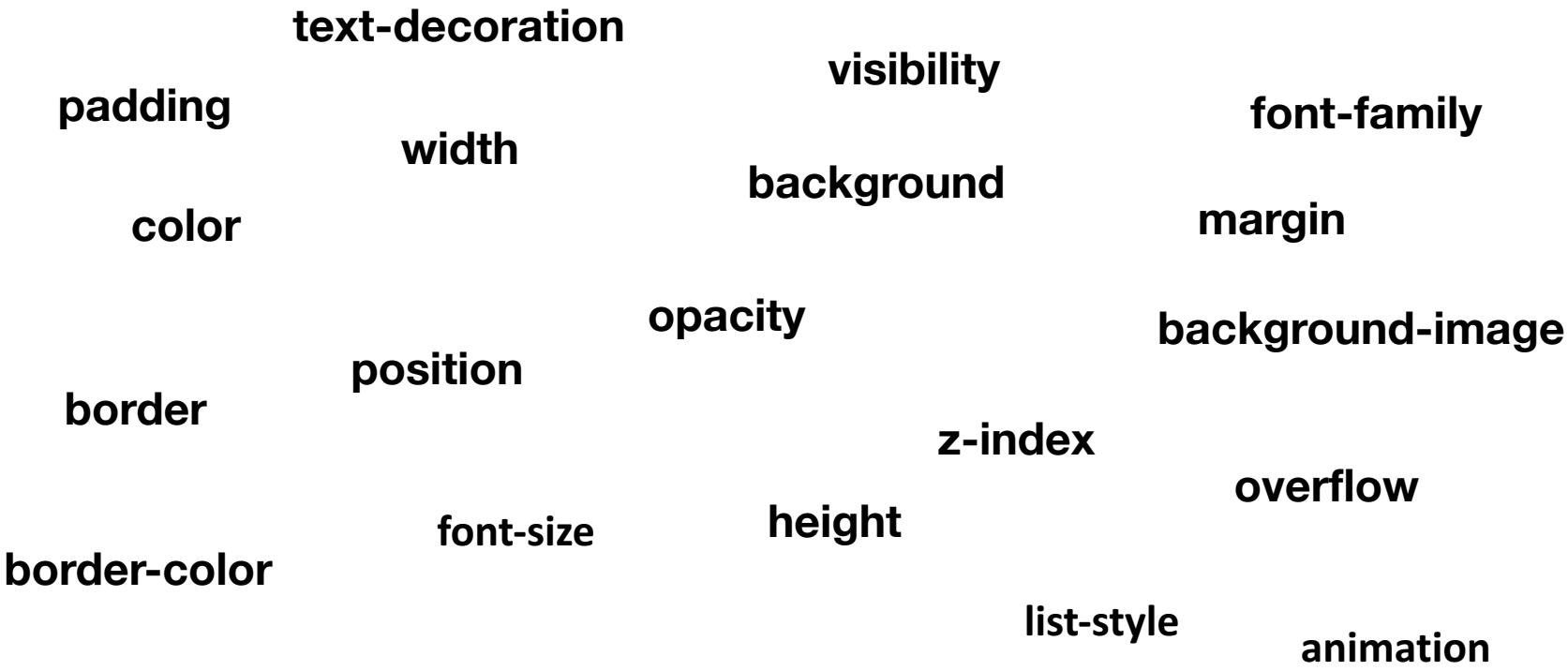
Changing **height** and **width**:

```
img {
    height: 200px;
    width: 200px;
}
```

Exercises:

1. Add the **above rules** to your index.html.
2. (Advanced) How can we make sure that the image keeps its **aspect ratio**?

Which CSS Styles Can We Apply? (cont'd)



Too many to memorize!

Use **online resources** for help:

- Google is your friend!
- http://www.w3schools.com/cssref/

Colors

There are a few **standard colors** you can use:

- Reference:
  http://www.w3schools.com/cssref/css_colornames.asp

But you can choose from more than 16 million colors by **mixing three components**: *red, green, and blue*.

- The amount for each component varies between 0 and 255 (0 means none of this color, 255 means all of this color).

255 0 0

0 255 0

0 0 255

```
body {
    background-color: rgb(199, 231, 220);
}
```

Fonts

- The **font-family** property is used to specify the font for an element.

```
p {
    font-family: Helvetica, "Trebuchet MS", sans-serif
}
```

- Not every font is supported by every browser/operating system combination.
- You can specify several font names as a “fallback” system.
- The browser tries to use the fonts in the order specified.
- There are commonly used font combinations: **Web Safe Fonts**.

These two sources are good starting points for more information:
http://www.w3schools.com/cssref/css_websafe_fonts.asp
https://www.google.com/fonts

CSS Selectors

- So far, we have only looked at the **type selector**.
  - The type selector selects elements based on the element name.

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

- There are **more powerful ways** to select elements in an HTML document, e.g.,:
  - ID selector
  - Class selector
  - Descendant selector
  - Nested selector

ID Selector:

HTML element IDs must be unique in a document!

- We can give HTML elements an **id**:

```
<p id="paragraph1">
    Some text.
</p>
```

- An id has to be **unique** in a HTML document.
- A CSS selector **preceded by a "#"** (hash) selects **one element** based on its id:

```
#paragraph1 {
    text-align: center;
    color: white;
}
```

Class Selector:

The same class name can appear multiple times in an HTML document!

- We can also give HTML elements a **class**:

```
<p class="priority">
    Some text.
</p>
```

- Multiple elements in a document can have the same class value.
- A CSS selector **preceded by a "."** selects **all elements** that belong to the class:

```
.priority {
    text-align: center;
    font-size: 45px;
}
```

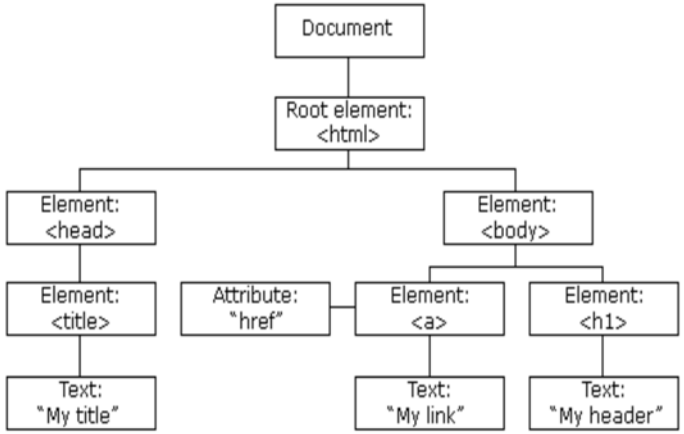
CSS Selectors (cont'd)

- **More Specific Class Selector:**
  - Selects all <p> elements with class "priority".

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

- **Descendant Selector:**
  - Selects all the paragraphs within a div.

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



There are more types of CSS Selectors. A good reference is http://www.w3schools.com/cssref/css_selectors.asp.

Section Recap: What have we learned?

SUMMARY

- What is CSS and why is it useful? ✓
- How do CSS and HTML work together? ✓
- What is a CSS rule? ✓
- Which CSS rule can we apply? ✓
- Fonts & Colors ✓
- What is a CSS selector and how do we use it for styling? ✓

MORE INFO

- Getting Started with CSS Selectors
  https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_started/Selectors

“Web design is responsive design.
Responsive web design is web design, done right.”

Andy Clarke, Designer, Speaker, Author

RESPONSIVENESS

OVERVIEW

- What does it mean for a website to be responsive?
- How can we make our website responsive?
- What is Bootstrap and how do we use it to build responsive web pages?

Responsive Web Design

How do we make sure that our web pages look professional **on any computer, tablet, mobile phone, and wearable**?

We have to make our website **responsive**!

1.	desktop	(58.06%)
2.	mobile	(39.78%)
3.	tablet	(2.15%)

Devices visiting importclasses.com

Another (related) issue:
How do we make sure that our web pages are correctly displayed in different browsers?

A responsive website **adapts its design** to **individual device requirements**.

For example, on a **phone**, users would see content shown in a single column view. A **tablet** might show the same content in two columns.

Some **basic guidelines** have been established:

- Do not use large **fixed width** elements.

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

```
div.largeContainer {
    width: 680px;
}
```

- Use **CSS media queries** to apply different styling depending on screen size.

```
@media screen and (min-width: 640px) {
    body {
        background-color: green;
    }
}
```

- Set **Viewport**.

More information on responsive web design can be found here:
https://developers.google.com/web/fundamentals/design-and-ui/responsive/fundamentals/

Bootstrap

- Bootstrap is the **most popular HTML, CSS, and JavaScript framework** for developing responsive websites.

- It is **free**, **open-source**, and **makes life easier**!
- Bootstrap contains **HTML** and **CSS templates** for **typography**, **forms**, **buttons**, **navigation**, etc.
- It uses a simple **grid-based system** that **resizes content automatically** based on device requirements.

Bootstrap examples:
http://expo.getbootstrap.com/

Bootstrap (cont'd)

- We can **integrate Bootstrap** directly into our website.
- To get the **Bootstrap CSS styles**, we simply add the following to our <head> section:

```
<head>
    <link rel="stylesheet" href="https://
    maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/
    bootstrap.min.css">
</head>
```

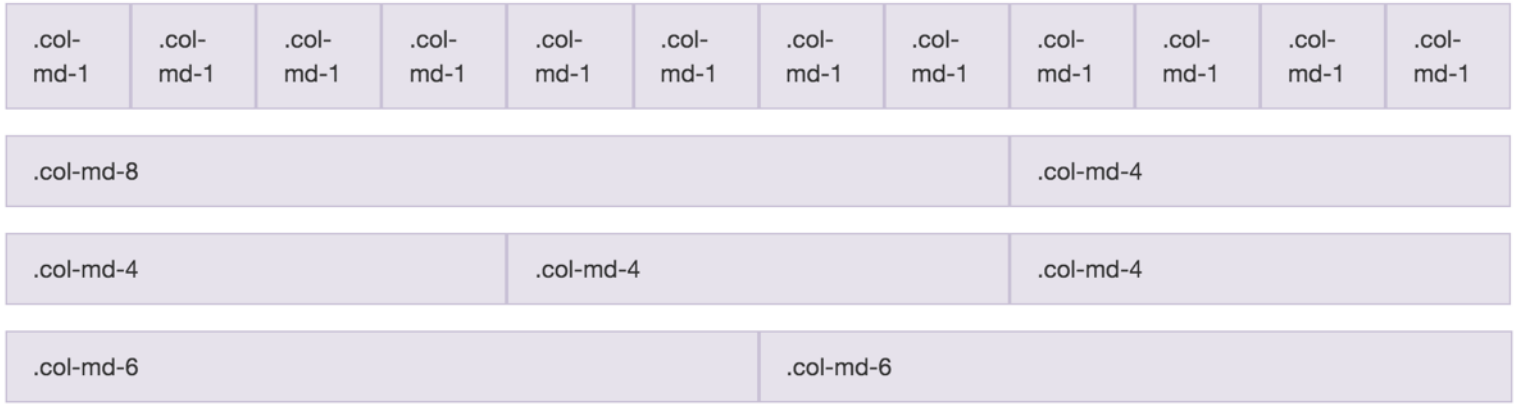
If you want to avoid typing the long URL, head over to http://getbootstrap.com/getting-started/ and copy it from there.

Exercise:

1. Add the above link tag to your **header section** in **index.html**.

Bootstrap Grid System

- Bootstrap's grid system uses **rows** and **columns** to to structure content.
- Depending on screen size, columns and rows will be **automatically scaled**.
- Columns are **stacked** on **mobile/tablet** screens and become **horizontal** on **desktop** devices.



Source: getbootstrap.com

- We use ** ** elements for layouting:

Rows are placed in a div with class "container".

```
<div class="container">
    <div class="row">
        <div class="col-md-4">I am column 1.</div>
        <div class="col-md-4">I am column 2.</div>
        <div class="col-md-4">I am column 3.</div>
    </div>
    <div class="row">
        <div class="col-md-8">I am a larger column.</div>
        <div class="col-md-2">I am a smaller column.</div>
        <div class="col-md-2">I am a smaller column.</div>
    </div>
</div>
```

Each row is a div with class "row".

col - md - 4

Specifies column

On which devices should columns be stacked?

Column size

Column sizes sum up to 12 in a row.

Adding Components to Your Website

- We can add a **<header> element** to our website with the following HTML code:

```
<header>
    <div class="container">
        <div class="jumbotron">
            <h1>A great title!</h1>
            <p class="lead">Some description.
            </p>
            <p>
                <a class="btn btn-lg btn-success" href="#"
                    role="button">See more</a>
            </p>
        </div>
    </div>
</header>
```

Exercises:

1. Add a **header section** to the top of the <body> in your HTML document.
2. How can we **center** the text in the header?

Adding Components to Your Website (cont'd)

- We can add a **navigation bar** to our website with the following HTML code above the <header>:

```
<nav class="navbar navbar-inverse navbar-fixed-top">
     <div class="container">
          <div class="navbar-header">
               <button type="button" class="navbar-toggle collapsed" data-
               toggle="collapse" data-target="#navbar" aria-expanded="false" aria-
               controls="navbar">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
               </button>
               <a class="navbar-brand" href="#">Awesome Project</a>
          </div>
          <div id="navbar" class="collapse navbar-collapse">
               <ul class="nav navbar-nav">
                    <li class="active"><a href="#">Home</a></li>
                    <li><a href="#overview">Overview</a></li>
                    <li><a href="#features">Features</a></li>
               </ul>
          </div><!--/.nav-collapse -->
     </div>
</nav>
```

You can also get the code if you type the following into your browser:
"view-source:http://getbootstrap.com/examples/starter-template/"

SUMMARY

- What does it mean for a website to be responsive? ✓
- How can we make our website responsive? ✓
- What is Bootstrap and how do we use it to build responsive web pages? ✓

MORE INFO

- Bootstrap Download and Documentation:
  http://getbootstrap.com
- Google Page Speed Insights
  https://developers.google.com/speed/pagespeed/insights/

“Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it.”

Brian Kernighan, Bell Labs Computer Scientist

INTRODUCTION TO JAVASCRIPT

OVERVIEW

- What is JavaScript and why is it useful?
- How do JavaScript, CSS, and HTML work together?
- How can we add interactivity to our website?
- How do we write JavaScript functions?
- What is jQuery, how does it relate to JS and why is it useful?

What is JavaScript?

HTML
to define the content and structure of your web page

CSS
for rich styling and layouting

JavaScript (JS)
to program the behavior of your websites

- JavaScript is a **high-level programming language**.
- The majority of websites employ JavaScript, and it is supported by all modern browsers.
- JavaScript runs in your browser (**client-side**).
- It is also used for server-side development, desktop and mobile applications, and game development.

JavaScript can:

- **change HTML content**
  We can modify the text in a paragraph when clicking on a button.

- **change HTML attributes and change CSS**
  We can hide and show elements (e.g., images, buttons) depending on the visitor's actions.

- **validate data forms**
  We can make sure that visitors complete registration forms and enter valid information.

Making Our Website Interactive

- We can add JavaScript to the **<head>** or **<body>.**
- If possible, it is **good practice** to **place JavaScript code at the bottom**, just before the closing </body> tag.

Exercises: First JavaScript Code

1. Add the following just **before** the **closing </body> tag** in index.html:

```
<script>
    alert("Hello, world.");
</script>
```

2. Reload the website and observe what happens.
3. Let's have JavaScript do some math for us. **Replace** the code from 1. with the following:

```
<script>
    var result = 5 + 3;
    alert("5 + 3 is " + result);
</script>
```

4. Reload the website.

Making Our Website Interactive (cont'd)

Additional Exercises: Make Some Text Appear

1. Add a **paragraph element** to your HTML document and give it **id "myParagraph"**. You do not have to add text to the paragraph for now but do not forget the closing tag!
2. Add a **button** by using the following code:

```
<button type="button">Click</button>
```

3. Reload the page and click on the button. What happens?
4. We can directly execute JavaScript code when this button is clicked. For this we need to add an **"onclick" attribute** to our button. Modify the button you added in 2. with the following code:

```
<button type="button"
onclick="document.getElementById('myParagraph').innerHTML =
'Hello World'">Click</button>
```

5. Reload the page and click on the button. What happens now?

Making Our Website Interactive (cont'd)

Additional Exercises: Changing Text Styling With JS Function

1. Add a **paragraph element** to your HTML document and give it **id "prettyParagraph"** and include some text of your choice.
2. Add a **button** below the paragraph element you added in 1. by using the following code:

```
<button type="button" onclick="applyStyling()">Make it pretty!</button>
```

3. Add the following JavaScript code **below the button** you added in 2.:

```
<script>
      function applyStyling() {
            var paragraph = document.getElementById("prettyParagraph");
            paragraph.style.color = "green";
            paragraph.style.fontSize = "29px";
            paragraph.style.fontWeight = "700";
      }
</script>
```

4. Reload the page and click the button to see what happens.

Making Our Website Interactive (cont'd)

Additional Exercises: Validating Form Input With JS (1/2)

1. Add an **input field** to your HTML document by using the following code:

```
<input id="numericInput" type="number">
```

2. Add an **empty paragraph element** with **id "result"** below the input field you added in 1.:

```
<p id="result"></p>
```

3. Add a **button** below the paragraph you added in 2. The button should have the title **"Test Input"** and call the JavaScript function **"testInput()",** which we will define in the next step.

Making Our Website Interactive (cont'd)

Additional Exercises: Validating Form Input With JS (2/2)

4. Add the following JavaScript code below the button you added in 3.:

```
<script>
      function testInput() {
            // Create variable to store test result
            var text;
            // Get the value of the input field with id="numb"
            var x = document.getElementById("numericInput").value;
            if (isNaN(x) || x < 0 || x > 99) {
                  text = "Wrong input";
            } else {
                  text = "Correct input";
            }
            // Display result
            document.getElementById("result").innerHTML = text;
      }
</script>
```

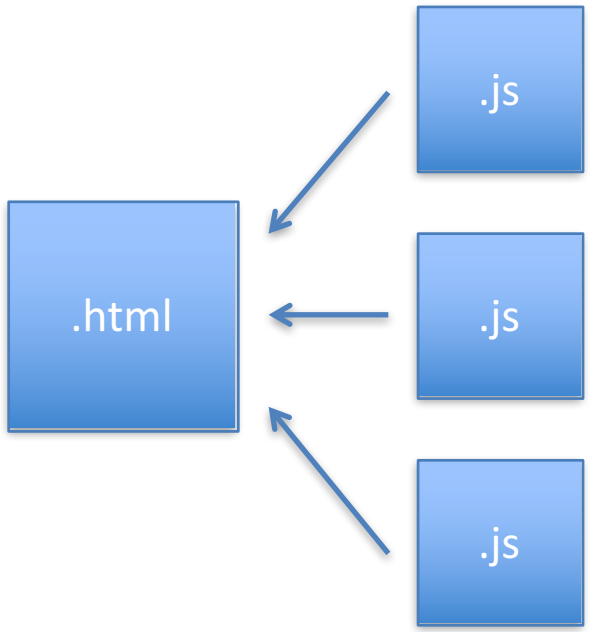
5. Reload the page. Enter and test a few different values to observe the page's behavior.

External JavaScript Files

- We can also load JavaScript code from an **external file**.
- JavaScript files end in **.js**.
- It can be **loaded** into our HTML file:

```
<script src="path/to/my/javascript/file.js"></script>
```

- **External JavaScript files** must **only contain JavaScript code**.
- External JavaScript files have **several advantages**:
  - It **separates** HTML and code.
  - It makes HTML and JavaScript **easier to read** and maintain.
  - Cached JavaScript files can **speed up** page loads.



- jQuery provides an **easier and more powerful way** to write JavaScript code.
- It is a free, open-source and widely-used **JavaScript library**.
- Before we can write code using jQuery we have to **import it in our HTML document**:

What does JS "library" mean?

*It's pretty much a collection JS code that we can reuse to do **more with less** code.*

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery
.min.js"></script>
```

Check jQuery.com for latest version!

Building an Animated HTML Carousel

- A **carousel** is a great way to **cycle through elements** on a web page.
- We need to first **include the Bootstrap JavaScript file** on our web page:

```
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstr
ap.min.js"></script>
```

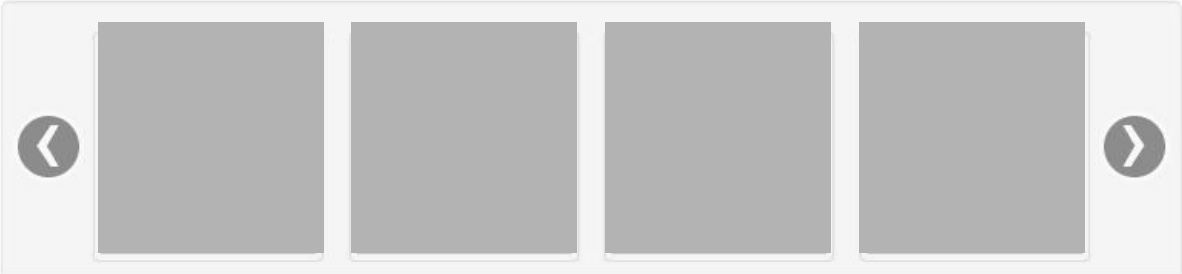
Exercises:

1. Add **jQuery** from the previous slide to the bottom of your HTML document, just **before the closing </body> tag**.
2. Add Bootstrap's **JavaScript** file to your HTML document. You can add it right **below jQuery**.

Building an Animated HTML Carousel (cont'd)

- We add the **carousel** structure with the following HTML code:

```
<div id="myCarousel" class="carousel slide" data-ride="carousel">
    <div class="carousel-inner" role="listbox">
        <div class="item active">
            I am the first element in the carousel.
        </div>
        <div class="item">
            I am the second element in the carousel.
        </div>
    </div>
</div>
```



Building an Animated HTML Carousel (cont'd)

- To **animate the carousel**, we need to add the following **jQuery code** to the bottom of our page:

This tells the browser that we are adding jQuery code (remember that jQuery is JavaScript code).

This function is called when the document has been loaded.

```
<script type="text/javascript">

    $(document).ready(function() {
        $("#myCarousel").carousel({interval: 4000});
    });

</script>
```

The carousel element we added on the previous slide has id *myCarousel*.

This animates the carousel. Every item is visible for four seconds.

Exercise:

1. Add the **carousel HTML code** from the previous slide to your HTML document. You can insert appropriate content for the respective items.
2. Add the **jQuery** code above to your HTML document.

Section Recap: What have we learned?

SUMMARY

- What is JavaScript and why is it useful? ✓
- How do JavaScript, CSS, and HTML work together? ✓
- How can we add interactivity to our website? ✓
- How do we write JavaScript functions? ✓
- What is jQuery, how does it relate to JS and why is it useful? ✓

MORE INFO

- JS Documentation and Tutorials:
  https://developer.mozilla.org/en-US/docs/Web/JavaScript
- jQuery Download and Documentation:
  https://jquery.com/

“The best method for accelerating a computer is the one that boosts it by 9.8 m/s^2.”

Anonymous

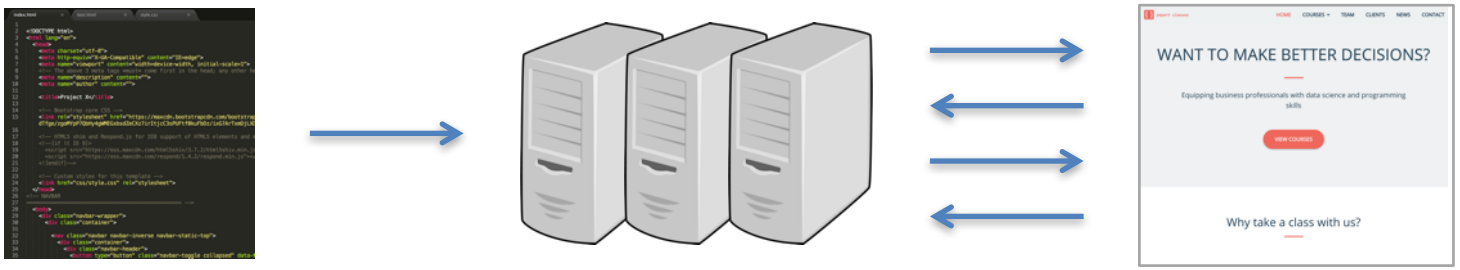
GO LIVE

OVERVIEW

- What is a Web Hosting Service?
- How do we make our website available on the internet?

Web Hosting Services

We require a **Web Hosting Service** that allows us to make our website **accessible via the World Wide Web**.



Your Website **Web Host** **Visitors**

Web Hosting Services (cont'd)

There is a **range of Web Hosting Services** that we can choose from.

What you might be **looking for**:

- Web Space
- Domain Name
- E-Mail Addresses
- Database Support

How do services **differ**?

- Features
- Support service
- Pricing
- Availability & Reliability
- Space

We can **host static websites** for free!

GitHub

HEROKU

Make sure you check out Github's Student Developer Pack which offers many tools/services for free (e.g., free domain for a year): https://education.github.com/pack

Hosting Static Websites (cont'd)

Exercise (1/2):

1. Go to **https://github.com/**. Create a new account if you do not already have a one.
2. Go to **https://github.com/new** to create a new public repository in your Github account. A good name for your repository would be "website".
3. Download install **git** on your computer. It is important to *keep the default settings* during the git installation process. You can find the download links here:
   - For **Windows**: *http://git-scm.com/download/win*
   - For **MacOS**: *http://git-scm.com/download/mac*
4. Use the *Command Prompt (Windows)* or *Terminal application (Mac OS)* to **navigate to the folder where you stored your website**.

Hosting Static Websites (cont'd)

Exercise (2/2):

5. Copy your **repository URL** from the GitHub page in your web browser:

6. Enter the following into your Command Prompt/Terminal:

```
git config --global user.name "<your name>"
git config --global user.email "<your email>"

git init
git add *
git commit -m "first commit"
git remote add origin <your repository URL>
git push -u origin master
```

7. Click on the **Settings** tab of your repository and scroll down to the **GitHub Pages** section. Then select the **master branch** source and click on the **Save** button.
8. Your website is **live** at *http://<username>.github.io/<repository name>.*

Section Recap: What have we learned?

SUMMARY

- What is a Web Hosting Service? ✓
- How do we make our website available on the internet? ✓

MORE INFO

- Github Pages
  https://pages.github.com/
- Heroku
  https://www.heroku.com/

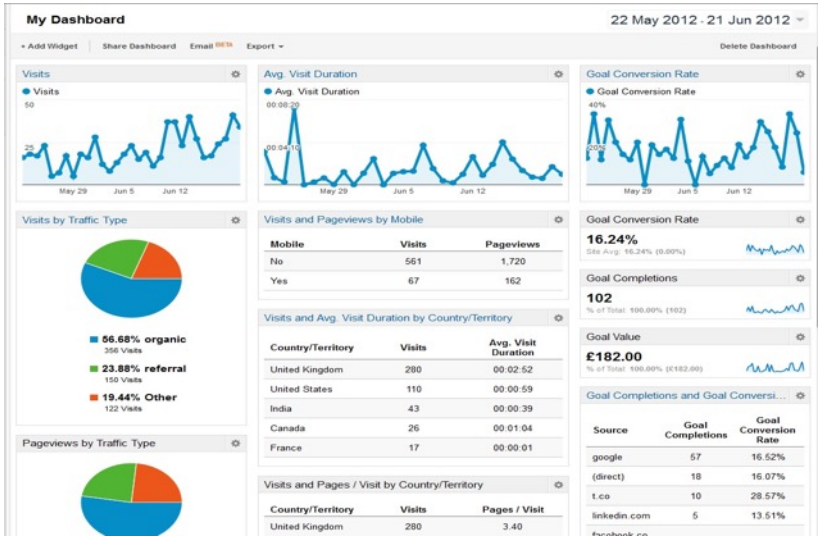
"Startup success can be engineered by following the process, which means it can be learned, which means it can be taught."

Eric Ries, Entrepreneur

MEASURE THE SUCCESS OF YOUR LANDING PAGE

OVERVIEW

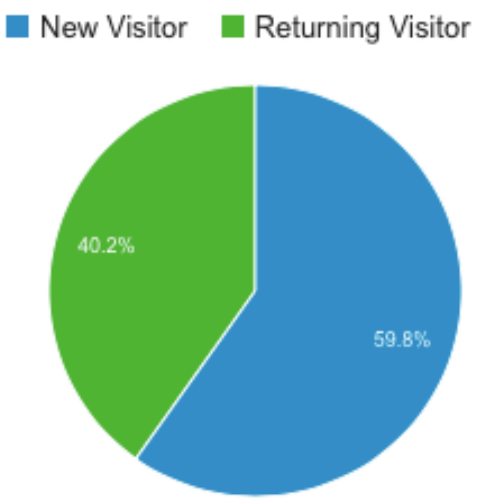
- Which insights can we get?
- What is Google Analytics and how can it help us?
- How can we integrate Google Analytics on our website?



Website Insights

There are **many questions** we can ask about our websites:

- How many people **visit** my website?
- What are the **characteristics** of my visitors?
- **Where** do my visitors come from?
- How much **time** do they spend on my website?
- What is the **typical behavior** of a visitor on my website?
- Which **devices and operating systems** do my visitors use when visiting our website?
- Which **traffic sources** are sending traffic to my website?
- How well does my website perform in regard to my **goals**?
- How **effective** is my website working?
- ...



***Lean Startup**: These questions tie in with the **"Measure" Phase** of the **Build-Measure-Learn Cycle**.*

Google Analytics

- **Google Analytics** is the most-widely used **web analytics service** on the internet.
- We can create a free account at *http://analytics.google.com*.
- Google Analytics uses a **JavaScript tracking code** that we have to add to our website. We integrate the following code just before the closing <head> tag:


```
<script type="text/javascript">
      var _gaq = _gaq || [];
      _gaq.push(['_setAccount', 'UA-XXXXXXXX-X']);
      _gaq.push(['_trackPageview']);

      (function() {
            var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async =
true;
            ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') +
                  '.google-analytics.com/ga.js';
            var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
       })();
</script>
```

We have to replace `'UA-XXXXXXXX-X'` with the tracking code for our Google Analytics account.

SUMMARY

- Which insights can we get? ✓
- What is Google Analytics and how can it help us? ✓
- How can we integrate Google Analytics on our website? ✓

MORE INFO

- Setting Up Google Analytics Web Tracking
  https://support.google.com/analytics/answer/1008080

WHAT HAVE WE LEARNED?

What Have We Learned?

- How the Internet Works ✓
- The Basics of HTML to structure our web pages ✓
- How we can integrate CSS and JavaScript to make our web pages look professional and interactive ✓
- What does it mean for a website to be responsive? ✓
- How we can make our website available to the world ✓
- How to use Google Analytics obtain insights into website user behavior ✓

Where To Go From Here

This course includes **post-workshop support** for three months after the event. You can **e-mail us** course-related questions to *support@cognitir.com* and we will get back to you within two business days.

To **stay up to date with our courses and free resource offerings**, sign up to our **newsletter** on *www.cognitir.com* and follow us on **social media**.

Please send **feedback** to *feedback@cognitir.com*.

 *@cognitir*

 *Cognitir*

 *Cognitir*

The journey of a thousand miles begins with one step.

Lao Tzu

CONGRATULATIONS!

“The goal as a company is to have customer service that is not just the best but legendary.”

Sam Walton, Founder of Walmart and Sam’s Club

Appendix

HTML FORMS

OVERVIEW

- What are HTML forms and what are they good for?
- How do we create HTML forms?
- How can we record and store information of our visitors?
- Create a signup form to record visitor information.

HTML **forms** are used to **collect user input**:

```
<form>
    First name:<br>
    <input type="text" name="firstname"><br>
    Last name:<br>
    <input type="text" name="lastname"><br>
    E-Mail:<br>
    <input type="text" name="email"><br>
    <input type="submit" value="Submit">
</form>
```

Exercises:

1. Add the above **form** to your HTML document.
2. Reload the page to see what the form looks like.
3. What happens if you click the **“Submit”** button?

How can we **store user input**?

We need a **database** to store user information. We can either submit information to our **own server/database** or use a **third-party service** for collecting user data.

MailChimp is an e-mail marketing service that lets us create forms which can be included on our websites. It also handles list management and newsletter campaigns.

Google Forms lets us collect and organize user information and store it in a spreadsheet.

Section Recap: What have we learned?

SUMMARY

- What are HTML forms and what are they good for? ✓
- How do we create HTML forms? ✓
- How can we record and store information of our visitors? ✓
- Create a signup form to record visitor information. ✓

MORE INFO

- W3 Schools on how to use HTML forms
  http://www.w3schools.com/html/html_forms.asp
- Mailchimp
  http://www.mailchimp.com
- Google Forms
  http://www.google.com/forms