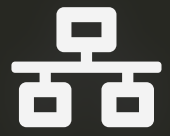
TWEB

Web Development

Bertil Chapuis

Overview of Today's Class

- Internet
- World Wide Web (WWW)
- Uniform Resource Locator (URL)
- HyperText Transfer Protocol (HTTP)
- Hypertext Markup Language (HTML)
- Cascading Style Sheets (CSS)



Internet

Internet's Conceptual Model *

The Internet Protocol Suite is the conceptual model and set of communications protocols used in the Internet and similar computer networks.

- The Application Layer specifies the shared communications protocols and interface methods used by hosts in a communications network.
  Examples: HTTP, HTTPS, FTP, SSH, SMTP, IMAP, Telnet, etc.

- The Transport Layer provide host-to-host communication services, such as connection-oriented communication, reliability, and flow control.
  Examples: TCP, UDP, etc.

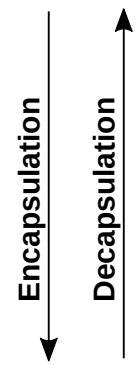
- The Internet Layer transports packets from the originating host across network boundaries to the destination host specified by an IP address.
  Examples: IP, ICMP (traceroute), IPsec (VPN), etc.

- The Link Layer is the group of methods and communications protocols that operate on the link that a host is physically connected to.
  Examples: ARP, PPP, MAC (Ethernet, Wifi, DSL, Fiber), etc.

* https://en.wikipedia.org/wiki/Internet_protocol_suite

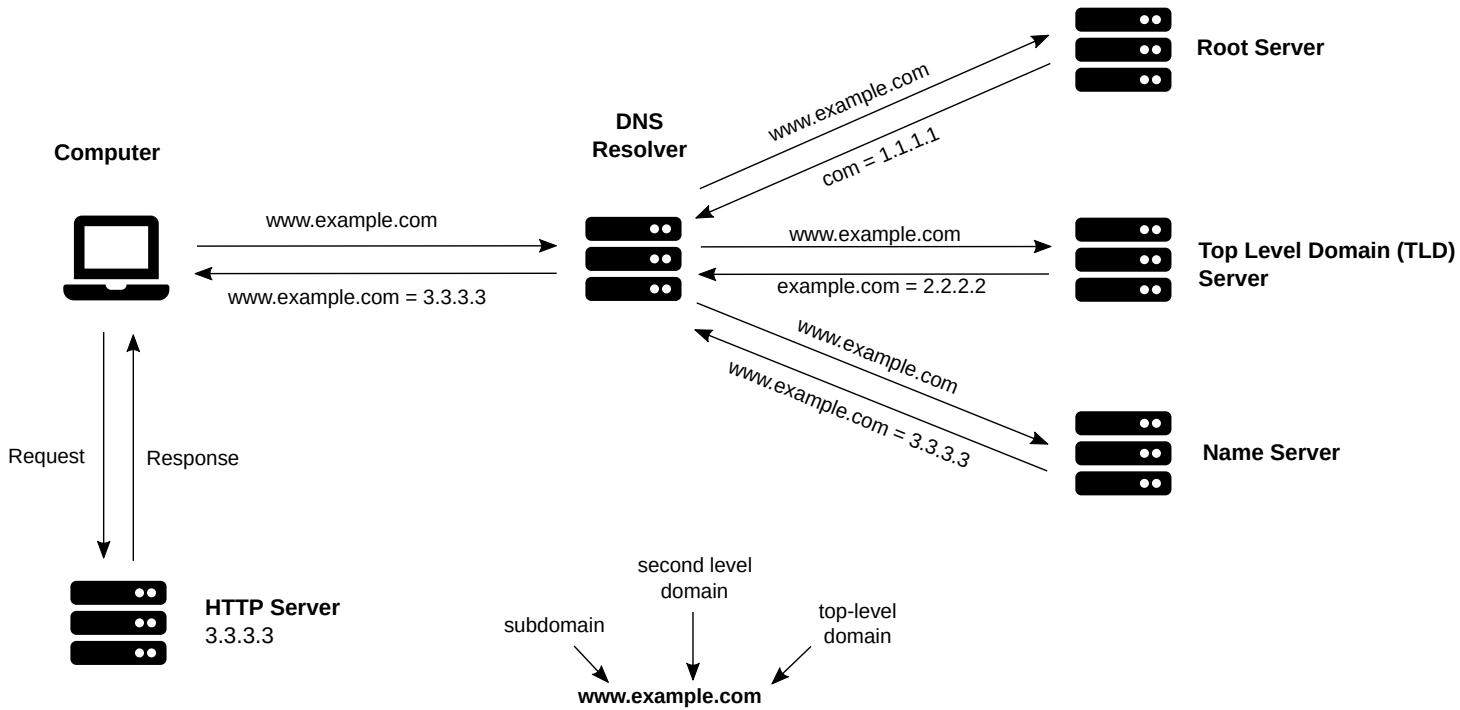
Internet's Conceptual Model

OSI	Internet Protocol Suite	Protocols	Data Unit
Application	**Application Layer**	**HTTP/HTTPS**	**Request/Response** req/rsp headers; etc.
Presentation			
Session			
Transport	**Transport Layer**	**SSL/TLS**	**Segment** src/dst port; seq num; ack; cheksum; etc.
		TCP	
Network	**Internet Layer**	**IP**	**Packet/Datagram** src/dst address; protocol; ttl; etc.
Datalink	**Link Layer**	**Ethernet/WiFi**	**Frame** sender/receiver mac; crc; etc.
Physical		**Wire/Fiber**	**Signal**



Domain Name System

The Domain Name System (DNS) is a hierarchical and decentralized naming system (phone book) for computers connected to the Internet. It translates domain names to IP addresses needed for locating and identifying computer.



The DNS protocol uses TCP for Zone transfer and UDP for name queries.

Zone file *

- A Domain Name System (DNS) zone file is a text file that describes a DNS zone.
- A DNS zone is a subset, often a single domain, of the hierarchical domain name structure of the DNS.
- The zone file contains mappings between domain names and IP addresses and other resources, organized in the form of text representations of resource records (RR).

Example

```
$ORIGIN example.com.                     ; start of this zone
$TTL 1h                                  ; default expiration time
example.com.  IN  MX    10 mail.example.com.  ; mailserver for example.com
example.com.  IN  A     192.0.2.1             ; IPv4 address for example.com
example.com.  IN  AAAA  2001:db8:10::1        ; IPv6 address for example.com
www           IN  CNAME example.com.          ; alias for example.com
```

* https://en.wikipedia.org/wiki/Zone_file

More about DNS

Mozilla provides a nice cartoon of how DNS works, what are its limitations in terms of security and privacy, and why DNS over HTTPS is needed.

https://hacks.mozilla.org/2018/05/a-cartoon-intro-to-dns-over-https/

Cloudflare provides a good introduction to DNS and how it is sometimes used to perform DNS amplification attacks and DNS flood attacks.

https://www.cloudflare.com/learning/ddos/glossary/domain-name-system-dns/

Hands on!

Perform some DNS lookups with the following commands:

```
nslookup -type=any heig-vd.ch
```

```
dig heig-vd.ch
```

Perform a reverse DNS lookup with the host command:

```
host wikipedia.org
```

```
host 91.198.174.192
```

Query the whois directory to check domain name ownership:

```
whois heig-vd.ch
```

Print the route packets trace to network host:

```
traceroute heig-vd.ch
```

World Wide Web

Mozilla's Definition *

The World Wide Web - commonly referred to as WWW, W3, or the Web - is an interconnected system of public webpages accessible through the Internet. The Web is not the same as the Internet: the Web is one of many applications built on top of the Internet.

The system we know today as "the Web" consists of several components:

- The HTTP protocol governs data transfer between a server and a client.
- To access a Web component, a client supplies a unique universal identifier, called a URL (uniform resource location) or URI (uniform resource identifier).
- HTML (hypertext markup language) is the most common format for publishing web documents.

* https://developer.mozilla.org/en-US/docs/Glossary/World_Wide_Web

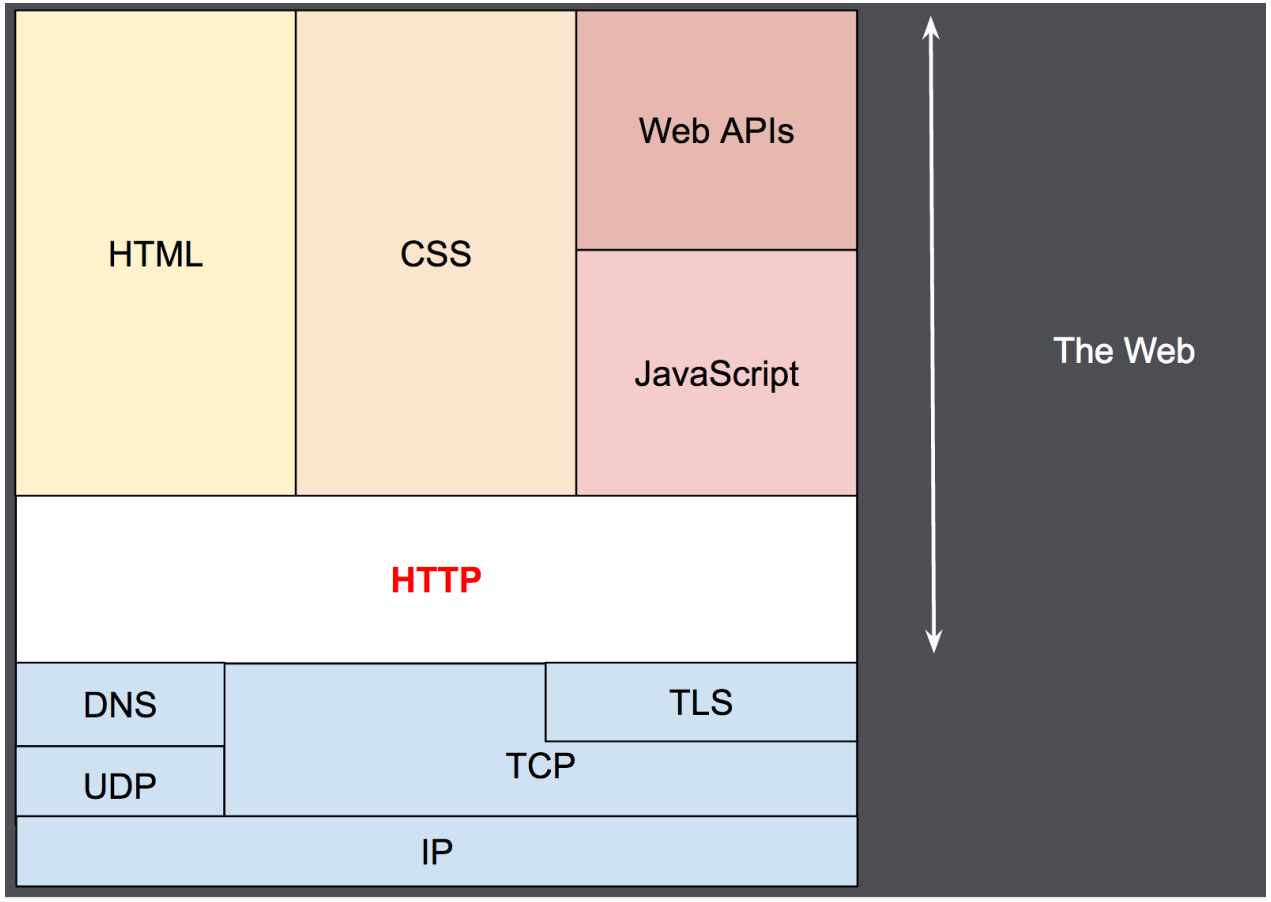
Mozilla's Definition *

Tim Berners-Lee proposed the architecture of what became known as the World Wide Web. He created the first web server, web browser, and webpage at the CERN in 1990. In 1991, he announced his creation, marking the moment the Web was first made public.

Today, the Web is constently evolving under the guidance of the World Wide Web Consortium (W3C).

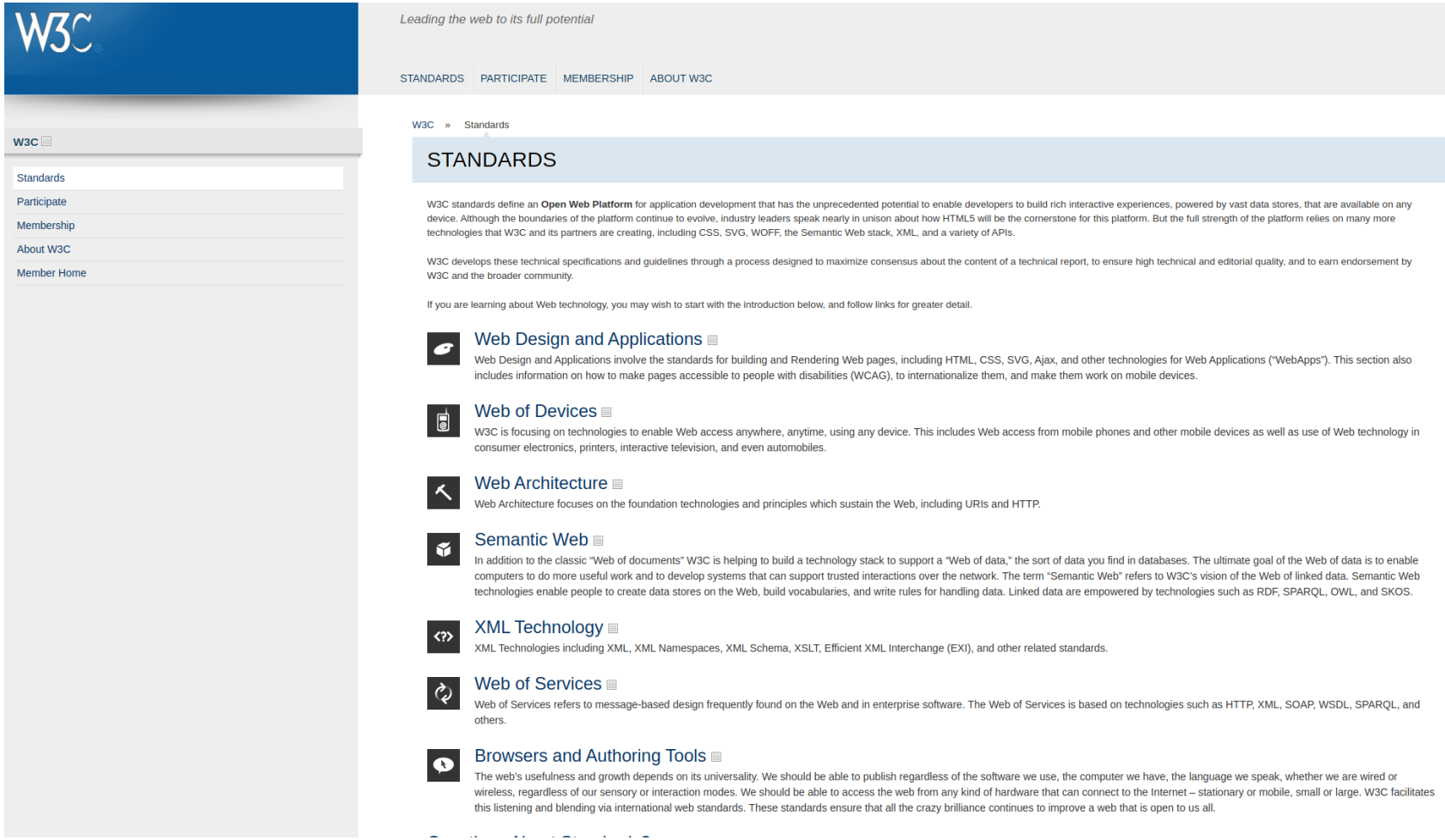
* https://worldwideweb.cern.ch/

Mozilla's Definition *



* https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

W3C's Standards



Mozilla's Web APIs

A

Ambient Light Events

B

Background Tasks

Battery API

Beacon

Bluetooth API

Broadcast Channel API

C

CSS Counter Styles

CSS Font Loading API

CSSOM

Canvas API

Channel Messaging API

Console API

Credential Management API

D

DOM

E

Encoding API

Encrypted Media Extensions

F

Fetch API

File System API

Frame Timing API

Fullscreen API

G

Gamepad API

H

HTML Drag and Drop API

High Resolution Time

I

Image Capture API

IndexedDB

Intersection Observer API

L

Long Tasks API

M

Media Capabilities API

Media Capture and Streams

Media Session API

Media Source Extensions

MediaStream Recording

N

Navigation Timing

Network Information API

P

Page Visibility API

Payment Request API

Performance API

Performance Timeline API

Permissions API

Pointer Events

Pointer Lock API

Proximity Events

Push API

R

Resize Observer API

Resource Timing API

S

Server Sent Events

Service Workers API

Storage

Storage Access API

Streams

T

Touch Events

V

Vibration API

W

Web Animations

Web Audio API

Web Authentication API

Web Crypto API

Web Notifications

Web Storage API

Web Workers API

WebGL

WebRTC

WebVR API

WebVTT

Websockets API

Uniform Resource Locator

What's in an URL?

A Uniform Resource Locator (URL), colloquially termed a web address, is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it.

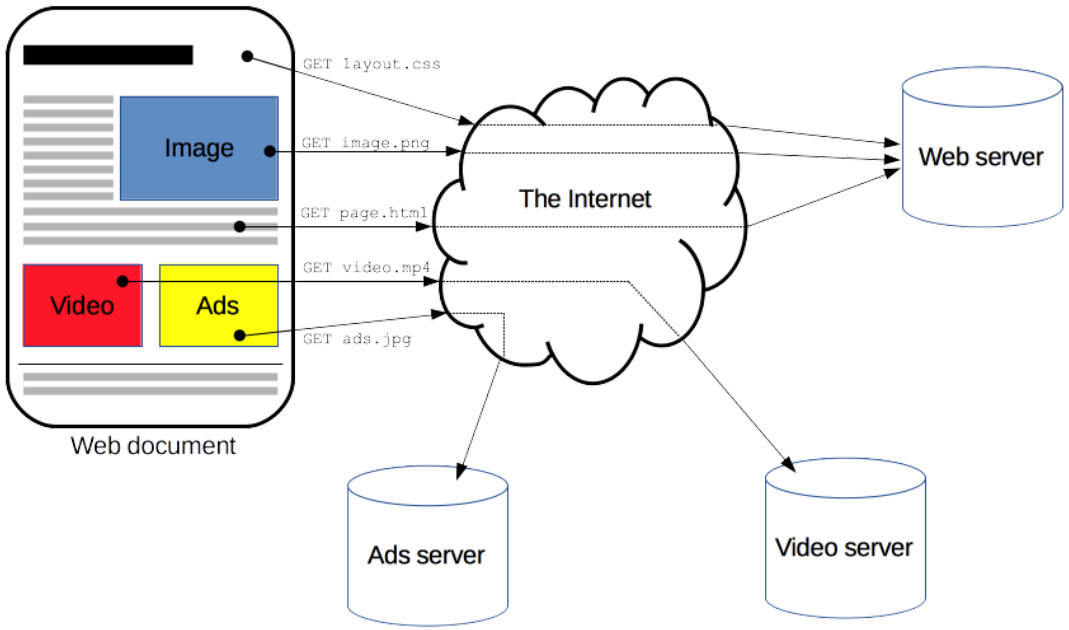
```
https://username:password@example.com:443/index.html?param=value#fragment
```

Part	Value	Description
Scheme	`https://`	The protocol to use for the request.
Credentials	`username:password@`	The credentials to use for the request (Basic Auth).
Domain	`example.com`	The domain name where to send the request.
Port	`:443`	The port of service endpoint.
Path	`/index.html`	The path of the resource.
Query	`?param=value`	The parameters associated with the resource.
Fragment	`#fragment`	The path of a secondary resource.

HTTP

Mozilla's Definition *

HTTP is a protocol which allows the fetching of resources, such as HTML documents. It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser. A complete document is reconstructed from the different sub-documents fetched, for instance text, layout description, images, videos, scripts, and more.



* https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

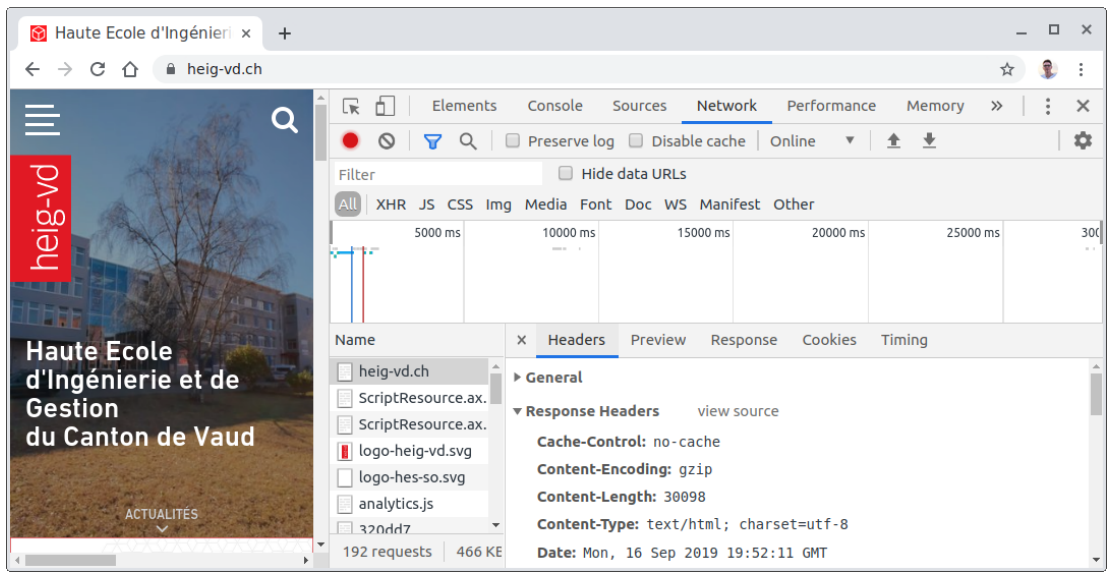
Hands on!

HTTP requests and responses are easy to look at!

Get a try with CURL:

```
curl -v http://httpstat.us/200?param=value
```

Or have a look at the DevTools in Chrome (CTRL+SHIFT+I):



Hands on!

Maybe too easy to look at…

Credentials and tokens can be captured by eavesdropping:

```
tcpdump -vvvs 1024 -l -A host www.heig-vd.ch \
  | strings \
  | grep -i "Authorization: Basic"
```

Now, what happen when you run the following command?

```
curl http://username:password@www.heig-vd.ch
```

HTTP Requests

```
GET /200?param=value HTTP/1.1
Host: httpstat.us
User-Agent: curl/7.58.0
Accept: */*
```

Requests usually have:

- a method (`GET`)
- a resource (`/200?param=value`)
- some headers (e.g. `User-Agent: curl/7.58.0`)
- an optional body (depends on the methods)

The most common Methods are:

- `GET`: Returns the resource.
- `POST`: Create resource.
- `HEAD`: Returns the headers of resource.
- `PUT`: Create or update resource.
- `DELETE`: Deletes resource:

HTTP Responses

```
HTTP/1.1 200 OK
Content-Length: 6
Content-Type: text/plain; charset=utf-8
Server: Microsoft-IIS/10.0
Access-Control-Allow-Origin: *
Date: Mon, 16 Sep 2019 20:07:29 GMT
200 OK
```

Responses usually have:

- a status code (`200 OK`)
- some headers (e.g. `Content-Length: 6`)
- an optional body (text, HTML, json)

The most common Status Codes are:

- `200 OK`: The request has succeeded (2xx Success).
- `301 Moved Permanently`: The resource has a new location (3xx Redirection).
- `404 Not Found`: The server has not found the resource (4xx Client Error).
- `500 Internal Server Error`: The server has not found the resource (5xx Server).

✋ Hands on!

Get to know your methods and status codes!

What is status code `418`?

✋ Hands on!

Get to know your methods and status codes!

What is status code `418`?

https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/418

`418 I'm a teapot`

The HTTP 418 I'm a teapot client error response code indicates that the server refuses to brew coffee because it is a teapot. This error is a reference to Hyper Text Coffee Pot Control Protocol which was an April Fools' joke in 1998.

by Julia Evans

https://gumroad.com/l/http-zine/buy-one-give-one

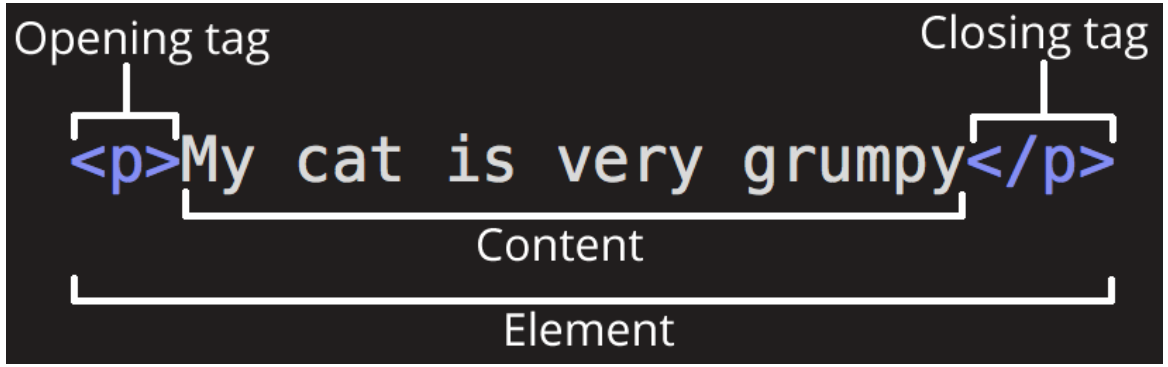

HTML

</> What is HTML? *

- HTML sands for **Hypertext Markup Language**
- HTML is the most basic **building block** of the Web
- HTML defines the **structure** of web content
- **Hypertext** refers to **links* that connect web pages to one another
- HTML uses **markup** to annotate text, images, and other content
- HTML5 is called a living standard as it is constantly evolving.

* https://developer.mozilla.org/en-US/docs/Web/HTML

Anatomy of an HTML element



An HTML element:

- starts with an opening tag
- may have some content
- stops with a closing tag

Attributes of an element

Attributes contain extra information about the element which you don't want to appear in the actual content.

- the `id` attribute must contains a unique value accross the document
- the `class` attribute usually refers to a class in a stylesheet
- the `style` attribute usually contains CSS properties

Anatomy of an HTML document

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My test page</title>
  </head>
  <body>
    <p>This is my page</p>
  </body>
</html>
```

- `DOCTYPE` needs to be included for everything to work right (historical)
- `<html>` wraps all the content on the entire page
- `<head>` is container for the stuff that isn't the content
- `<meta charset="utf-8">` sets the character set your document should
- `<title>` sets the title of the web page.
- `<body>` contains all the content that you want to show to web users
- Elements can be nested

Text elements

Headings:

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
```

Paragraph

```
<p>Paragraph</p>
```

Line break:

```
<br />
```

Horizontal line:

```
<hr />
```

Semantic elements

Header (introductory content):

```
<header></header>
```

Main (the dominant content of the body):

```
<main></main>
```

Footer:

```
<footer></footer>
```

Section elements

Division (content block):

```
<div></div>
```

Navigation:

```
<nav></nav>
```

Article (piece of self-contained content):

```
<article></article>
```

Section (grouping of semantic meaning):

```
<section></section>
```

Aside (content that does not belong to the main content):

```
<aside></aside>
```

Hyperlink element

```
<a href="https://www.heig-vd.ch" title="Heig-vd" target="_blank">Heig-vd</a>
```

```
<a href="file:///home/bchapuis/Projects/github.com/tweb/slides/mailto:username@email.com?subject=hello&body=world!">
```

The hyperlink element must contain an `href` attribute and can specify a `title` and a `target`. The content of the hyperlink can be an image.

Image element

```
<img src="undefined" alt="caption" />
```

The image element must have an `src` (image url) and `alt` (caption of the image) attribute.

Audio element

```
<audio src="file:///home/bchapuis/Projects/github.com/tweb/slides/audio.mp3"></audio>
```

Canvas element

```
<canvas></canvas>
```

The HTML canvas allows for dynamic, scriptable rendering of 2D and 3D shapes.

Nested lists

Numbered lists and enumerations can be nested.

```
<ol>
  <li>A</li>
  <li>B
    <ul>
      <li>C</li>
      <li>D</li>
      <li>E</li>
    </ul>
  </li>
  <li>F</li>
</ol>
```

Tables

Tables should only be used for tabular data.

```
<table>
  <tr><!-- row -->
    <th>Student ID</th> <!-- header column -->
    <th>Grade</th>
  </tr>
  <tr>
    <td>4</td> <!-- column -->
    <td>5</td>
  </tr>
  <tr>
    <td>6</td>
    <td>7</td>
  </tr>
</table>
```

The `colspan` and `rowspan` attributes can be used to merge cells.

Why using meta tags?

Meta tags are a great way for webmasters to provide search engines with information about their sites.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="Description" content="Great description" />
```

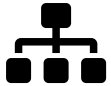
https://support.google.com/webmasters/answer/79812?hl=en

HTML Entities

An HTML entity is a piece of text ("string") that begins with an ampersand (&) and ends with a semicolon (;) . Entities are frequently used to display reserved characters.

Character	Entity	Note
&	`&`	Interpreted as the beginning of an entity or character reference.
<	`<`	Interpreted as the beginning of a tag
>	`>`	Interpreted as the ending of a tag
"	`&quot;`	Interpreted as the beginning and end of an attribute's value.

https://developer.mozilla.org/en-US/docs/Glossary/Entity



What is the DOM? *

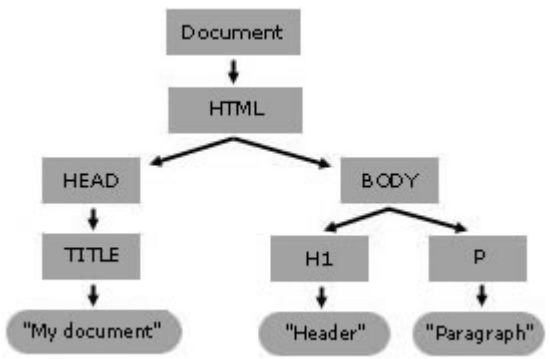
- DOM stands for **Document Object Model**
- The DOM is a programming interface for HTML and XML
- The DOM represents the structure of a document in memory
- It lets other programming languages manipulating the document

* https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model

The DOM's content tree *

```
<html>
<head>
  <title>My Document</title>
</head>
<body>
  <h1>Header</h1>
  <p>Paragraph</p>
</body>
</html>
```

When a browser such as Chrome or Firefox parses an HTML document, it builds a **content tree** and then uses it to **display** the document.



* https://developer.mozilla.org/en-US/docs/Web/API/Document_object_model/Using_the_W3C_DOM_Level_1_Core


CSS

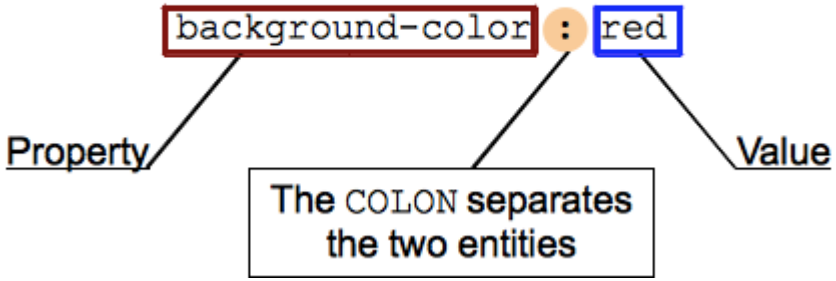
What is CSS? *

- CSS stands for **Cascading Style Sheets**
- CSS is a core language of the **open Web**
- CSS is a **stylesheet language** for HTML or XML documents (including XML dialects such as SVG, MathML or XHTML)
- CSS describes how **elements** should be **rendered** on a media (a media can be a screen, a paper, etc.)
- CSS is **standardized** across Web browsers according to the W3C specification
- CSS3 is the latest version of the standard.

* https://developer.mozilla.org/en-US/docs/Web/CSS

CSS declaration *

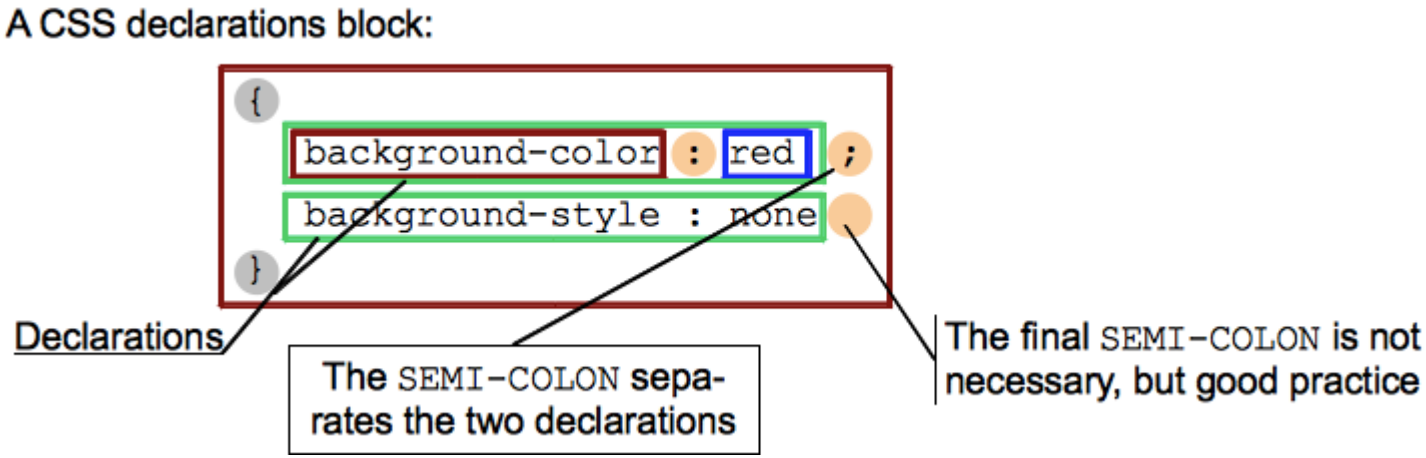
A CSS declaration :



- Setting **CSS properties** is the core function of the CSS language
- A **property** and **value** pair is called a **declaration**
- **Properties** and **values** are case-insensitive
- The pair is separated by a colon `:`
- There are more than 100 different properties in CSS

* https://developer.mozilla.org/en-US/docs/Web/CSS/Syntax

CSS declarations block *

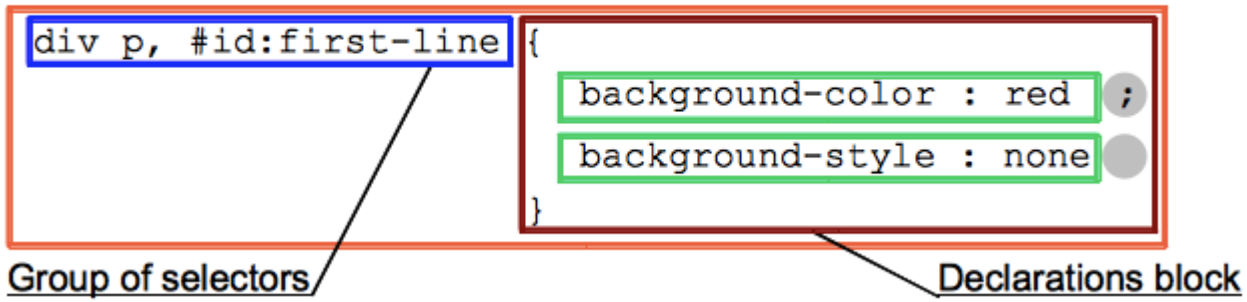


- CSS declarations are grouped in **blocks**
- CSS declarations are **separated** by a **semi-colon** (`;`)
- Blocks are **delimited** by an opening (`{`) and a closing **brace** (`}`)
- Sometimes blocks can be **nested** (e.g., media queries)

* https://developer.mozilla.org/en-US/docs/Web/CSS/Syntax

CSS Rulesets *

A CSS ruleset (or rule):



- **Rulesets** apply declarations to **specific** parts of the document
- **Declaration blocks** are preceded by one or more comma-separated **selectors**
- **Selectors** are conditions selecting some elements of the page
- **Cascading** refers to the precedence of the **selectors** over each others

* https://developer.mozilla.org/en-US/docs/Web/CSS/Syntax

Adding Stylesheet to an HTML Document

Stylesheets can be **embed** in the HTML document:

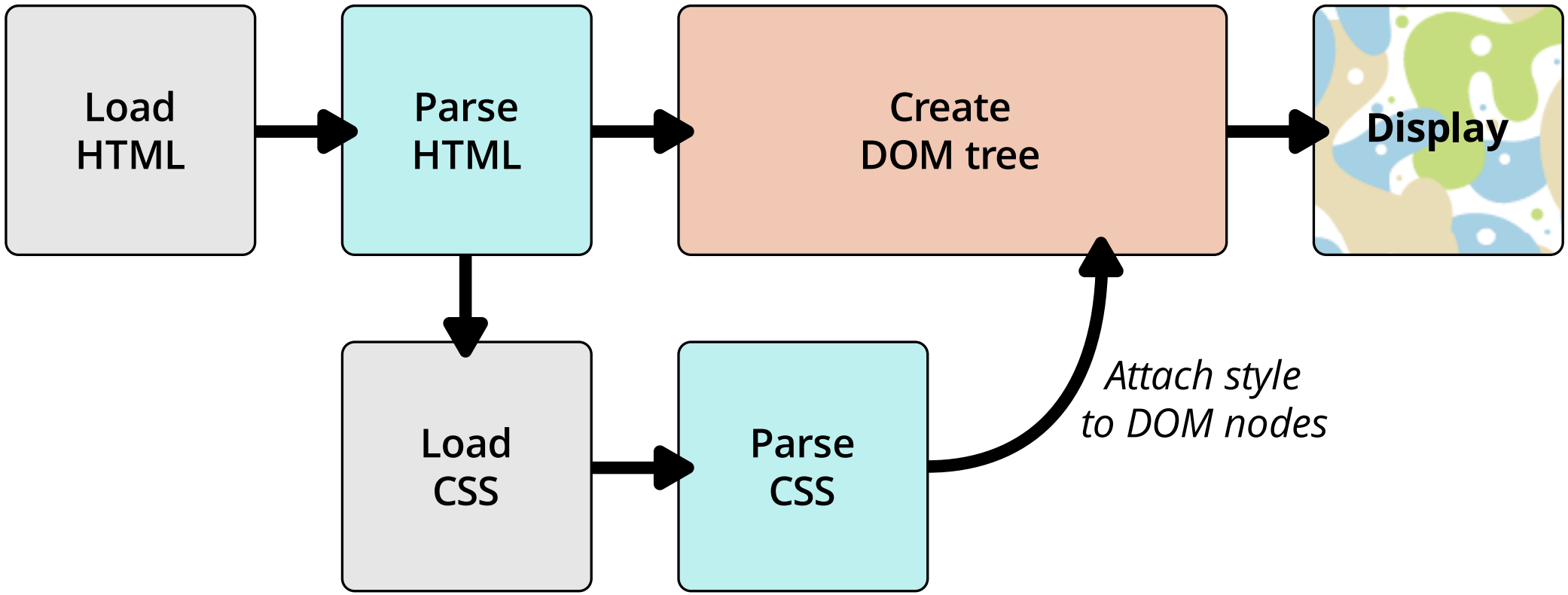
```
!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-color: linen;
      }
    </style>
  </head>
  <body></body>
</html>
```

Stylesheets are usualy stored in **external** files:

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="file:///home/bchapuis/Projects/github.com/tweb/slides/style.css">
  </head>
  <body></body>
</html>
```

How does CSS work?*

When a browser displays a document, it must combine the document's content with its style information.



* https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/How_CSS_works

CSS Selectors *

The **type selector** selects the elements that match the given node name:

```
p {}
```

The **id selector** selects the elements that have a given id attribute:

```
#myid {}
```

The **class selector** selects the elements that have a given class attribute:

```
.myclass {}
```

The **universal selector** select all the elements:

```
* {}
```

The **attribute selector** select the elements with a given attribute:

```
[attr=value] {}
```

* https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors

CSS Combinators *

CSS Combinators can be used to mix several selectors.

The (space) combinator selects nodes that are descendants of the first element.

```
ul li {}
```

The `>` combinator selects nodes that are direct children of the first element.

```
ul > li {}
```

Other combinators, such as `+`, `~`, respectively applies to **adjacent** and **sibling** elements in the DOM.

* https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors

CSS Pseudo-class *

A **pseudo-class** is a keyword added to a selector that specifies a special state of the selected element(s).

The `:hover`, `:link`, `:visited` and `:active` pseudo-class matches when the user interacts with an element with a pointing device.

```
a:hover {background-color: red; }
```

The :nth-child() CSS pseudo-class matches elements based on their position in a group.

```
td:nth-child(2n) { background-color: gray; }
```

* https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes

More Pseudo-classes

:active
:any-link
:blank
:checked
:current
:default
:defined
:dir()
:disabled
:drop
:empty
:enabled
:first
:first-child
:first-of-type
:fullscreen
:future
:focus
:focus-visible
:focus-within
:has()
:host
:host()
:host-context()
:hover
:indeterminate
:in-range
:invalid
:is()
:lang()
:last-child
:last-of-type
:left
:link
:local-link
:not()
:nth-child()
:nth-col()
:nth-last-child()
:nth-last-col()
:nth-last-of-type()
:nth-of-type()
:only-child
:only-of-type
:optional
:out-of-range
:past
:placeholder-shown
:read-only
:read-write
:required
:right
:root
:scope
:target
:target-within
:user-invalid
:valid
:visited
:where()

Hands on!

Experiment with the **DOM** and with **selectors** in Boris's CSS Visualizer.

https://fritscher.ch/dom-css/

```
#abc
.abc
h1#abc
h1
h1[class=abc]
body > h1
ul li
ul li:nth-child(2)
```

Color Property *

The color property sets the foreground color value of an element's text.

```
p {
  /* named-color values */
  color: red;
  color: orange;

  /* hex-color values */
  color: #090;
  color: #009900;

  /* rgb() values */
  color: rgb(34, 12, 64, 0.6);
  color: rgba(34, 12, 64, 0.6);
}
```

Color values can be used with other CSS properties such as:

```
background-color: #090;
border-color: #090;
```

* https://developer.mozilla.org/en-US/docs/Web/CSS/color

Font Properties *

```
p {
  font-family: Times New Roman, serif, Arial, sans-serif, Consolas, monospace;
  font-style: italic;
  font-weight: bold;
  font-size: 1.5em;
}
```

The font property sets an element's font to a system font.

- The first font specified that is available is used to display the element.
- The recommended font size units are `px`, `em` and `%`. Pixels are fixed, wheareas `em` and `%` are relative to the font size specified for the document.

Additional fonts can be imported in a stylesheet with the import directive.

```
@import url(file:///home/bchapuis/Projects/github.com/tweb/slides/'https://fonts.googleapis.com/css?family=Roboto&di

body{
  font-family: 'Roboto', sans-serif;
}
```

fonts.google.com and fonts.com provide public registries for fonts.

* https://developer.mozilla.org/en-US/docs/Web/CSS/font

Text Properties *

```css
p {
  text-align: center;
  line-height: 2em;
  letter-spacing: 2em;
  text-decoration: underline;
  text-transform: uppercase;
}
```

Many CSS properties can be used to to perform text manipulation, like line breaking, justification and alignment, white space handling, and text transformation.

* https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Text

Background Properties *

```
body {
  background-color: blue;
  background-image: url(undefined);
  background-repeat: no-repeat;
  background-size: auto;
}
```

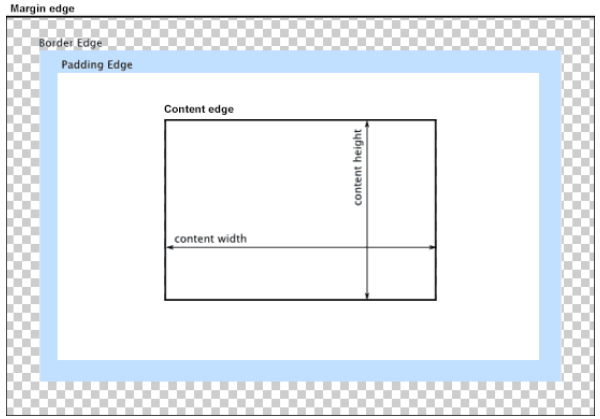
Many CSS properties applies to the background of an HTML element.

* https://developer.mozilla.org/en-US/docs/Web/CSS/background

Box Model *

```
div {
  display: block;
  margin: 20px 10px 20px 15px;
  padding: 10px;
  border: solid 1px red;
  border-radius: 20px;
  width: 200px;
  height: 100px;
}
```

The browser's rendering engine represents each element as a rectangular box according to the standard CSS basic box model.



* https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Box_Model/Introduction_to_the_CSS_box_model

Positionning *

```html
<div id="el">My element</div>
```

```css
#el {
  position: fixed
  top: 100px;
  left: 100px;
  bottom: 100px;
  right: 100px;
}
```

The position CSS property sets how an element is positioned in a document.

- `static`: The element is positioned according to the normal flow of the document.
- `relative`: The element is positioned according to the normal flow of the document, and then offset relative to itself.
- `fixed`: The element is positioned relative to its closest positioned ancestor.
- `absolute`: The element is positioned relative to the initial containing block established by the viewport.

* https://developer.mozilla.org/en-US/docs/Web/CSS/position

CSS Flexbox *

The flexbox model addresses the limits of the layout system (grid) in CSS.

```
<div class="container">
  <div class="item">Item A</div>
  <div class="item">Item B</div>
  <div class="item">Item C</div>
</div>
```

```
.container {
  display: flex;
  flex-direction: row;
}
.item {
  order: 1;
  flex-grow: 1
}
```

CSS tricks provides a very good tutorial on Flexbox.

* https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox

CSS Media Queries *

Media queries are useful when you want to modify your site or app depending on a device's general type (such as print vs. screen) or specific characteristics and parameters (such as screen resolution or browser viewport width).

```
@media only screen and (min-width : 600px) {
  body {
    color: red
  }
}
```

* https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries

Questions ?

Questions about Today's Lecture

- Internet
- World Wide Web (WWW)
- Uniform Resource Locator (URL)
- HyperText Transfer Protocol (HTTP)
- Hypertext Markup Language (HTML)
- Cascading Style Sheets (CSS)

Group Assignment

Group Assignment

- Form groups of max. 4 students
- Install Visual Studio Code, Node.js and Docker
- If needed, watch Olivier's webcast to setup your environment
- Go to the Github Classroom and start exercise 1 (HTML & CSS)
- **Interact** with the assistants if needed… ;)
- The repository will be frozen **next Tuesday at 12am**

 Questions ?