Web Technologies

Lab session 2

HTML5

- HTML: Hyper-Text **Markup** Language
  - Used to markup (annotate) the content
  - Not to define the outlook (design)
- All modern browsers *support* HTML5
  - Support for older browsers can be added manually
  - https://www.w3schools.com/html/html5_browsers.asp

HTML5: Novelties

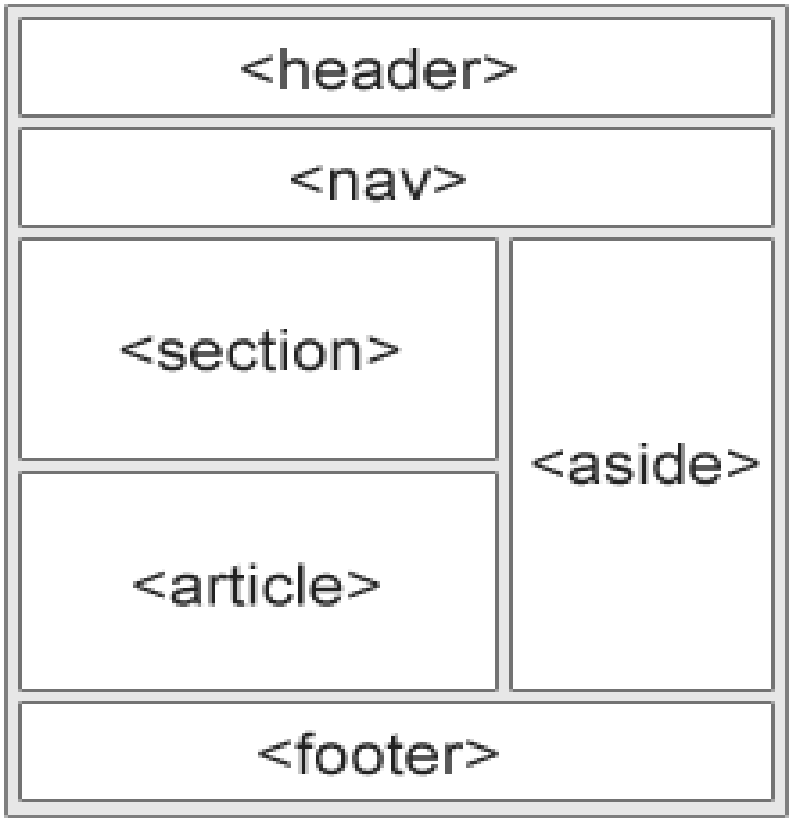
- Elements:
  - Semantic elements: header, footer, article, section
  - Attributes for form elements: number, date, time, calendar
  - Graphic elements: svg, canvas
  - Multimedia elements: audio, video
- API
  - HTML Geolocation
  - HTML Drag and Drop
  - HTML Local Storage
  - HTML Application Cache
  - HTML Web Workers
  - HTML SSE (server-sent events)

HTML5: Semantic elements

- Semantic elements = elements that have meaning
  - Easier for search engines and other software to parse web pages
- **section**: a thematic grouping of content, typically with a heading
- **article**: independent, self-contained content; should stand on its own; forum post, blog post, newspaper article
  - Ambiguity: Is section a part of an article or is an article a part of a section?
- **footer, header, nav, aside, details …**

https://www.w3schools.com/html/html5_semantic_elements.asp

HTML5: Semantic elements



https://www.w3schools.com/html/html5_semantic_elements.asp

CSS3

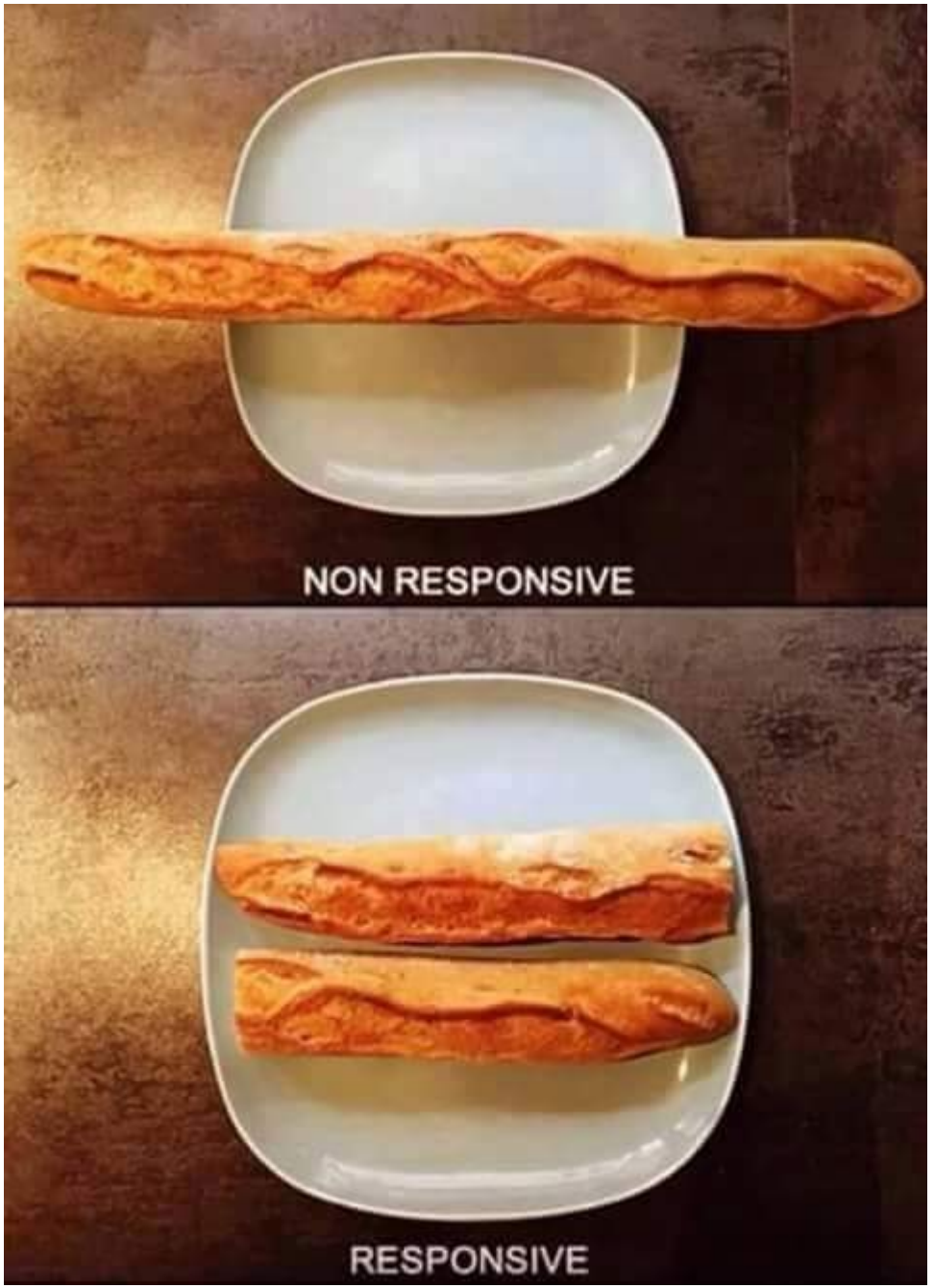
- Latest version of the CSS standard
- Some of the CSS3 modules:
  - Selectors, Box Model, Backgrounds and Borders, Image Values and Replaced Content, Text Effects, 2D/3D Transformations, Animations, …
  - https://www.w3schools.com/css
- Browsers implemented CSS3 ahead of specification and they have their own commands

```
.newspaper {
  -webkit-column-count: 3; /* Chrome, Safari, Opera */
  -moz-column-count: 3;    /* Firefox */
  column-count: 3;         /* The standardized command */
}
```

Responsive Web Design (RWD)

- Responsive web design
  - A design that adapts to the size of the screen
  - Uses only HTML and CSS, no JavaScript
- Web is not only for desktop: tablets, phones, and other devices
- Pages should adapt its content to fit any device

https://www.w3schools.com/css/css_rwd_intro.asp


NON RESPONSIVE
RESPONSIVE

RWD: Viewport

- Viewport: visible area of a web page: large on desktop, small on phones
- Use meta tag to set the viewport size and prevent initial zoom-out
- By default, smaller viewports will scale down the page (zoom it out) to fit the content; we do not want that

```
<meta name="viewport"
content="width=device-width, initial-scale=1.0">
```

https://www.w3schools.com/css/css_rwd_viewport.asp

RWD: Key principles

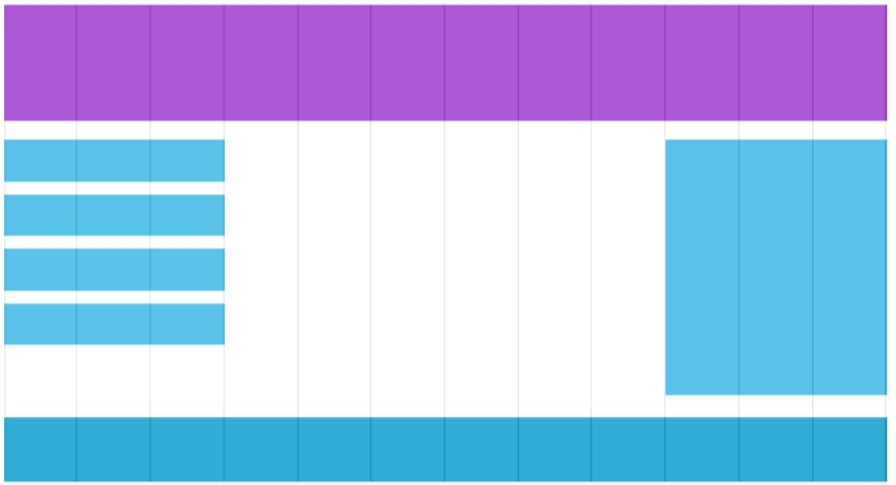
- **Flexible layouts** denote the practice of building the layout with a flexible grid, capable of dynamically resizing to any width
- **Media Queries** provide the ability to specify different styles for different browser and device configurations
- **Flexible Media** denotes the scalability of images, videos, and other media types as the size of the viewport changes

RWD: Implementation

- The “old way” using **floats**
- The CSS3 way using
  - **flex** layout, or
  - **grid** layout

RWD: Flexible grid: grid-view

- Often we use a **grid-view**, which means that the page is divided into columns
- Easy to place elements. Typically, we use 12 columns



Size of elements should be given in percentages.

RWD: Media queries (MQ)

- Media queries are part of CSS3
- Idea is to use **@media** rule to include a block of CSS properties only if a certain condition holds
  - https://www.w3schools.com/cssref/css3_pr_mediaquery.asp

```
@media not|only mediatype and (media feature) {
    CSS-Code;
}
```

- These rules are called **breakpoints**

RWD: Media queries (MQ)

- When the viewport is smaller than 500px, the background color changes to light blue

```
body {
    background-color: lightgreen;
}

@media only screen and (max-width: 500px) {
    body {
        background-color: lightblue;
    }
}
```

RWD: MQ: Mobile first

- Mobile first means using styles targeted at **smaller viewports as the default** and then using media queries to add styles for larger viewports
  - A mobile user should not have to load the styles for a desktop computer only to have them overwritten with mobile styles later
    - Waste of bandwidth and processing (battery life)
- Be **mobile friendly**: soon the majority of the web will be accessed from mobile devices

RWD: Flexible media

- Media (images video) should change with the viewport
  - small screens small media, larger screens larger media
- Easy fix

```
img, video, canvas {
  max-width: 100%;
  height: auto;
}
```

Assignment 1

- Implement a style for mobiles (widths up to 600px):
  - Each column should be of 100% width
- Implement a style for tablets (widths up to 992px)
  - The **nav** and **article** should be in the same row
  - The **aside** should be in new row
- Use the mobile first implementation strategy
- Add sunrise.png image and make it responsive
- https://www.w3schools.com/css/css_rwd_mediaqueries.asp

Lab session 2: Responsive Web Design

Overview

Responsive web design makes your web page look good on all devices.

Responsive web design uses only HTML and CSS.

Responsive web design is not a program or a JavaScript.

Three principles

For a design to be called responsive, it has to adhere to three principles: **flexible layouts**, **media queries**, and **flexible media.**

Flexible layouts

Flexible layouts denote the practice of building the layout with a flexible grid, capable of dynamically resizing to any width.

Media Queries

Media queries provide the ability to specify different styles for individual browser and device circumstances, the width of the viewport or device orientation for example.

Flexible Media

Images, videos, and other media types need to be scalable, changing their size as the size of the viewport changes.

Flexible layouts

Layout should be a flexible grid that dynamically resizes to any width.

Media Queries

Media queries allow defining different styles for different browser and device configurations.

Flexible Media

Images, videos, and other media types should also be dynamically resizable.

Lab session 2: Responsive Web Design

Index

HTML5: SVG

Item 3

Item 4

Overview

Responsive web design makes your web page look good on all devices.

Responsive web design uses only HTML and CSS.

Responsive web design is not a program or a JavaScript.

Three principles

For a design to be called responsive, it has to adhere to three principles: **flexible layouts**, **media queries**, and **flexible media.**

Flexible layouts

Flexible layouts denote the practice of building the layout with a flexible grid, capable of dynamically resizing to any width.

Media Queries

Media queries provide the ability to specify different styles for individual browser and device circumstances, the width of the viewport or device orientation for example.

Flexible Media

Images, videos, and other media types need to be scalable, changing their size as the size of the viewport changes.

Flexible layouts

Layout should be a flexible grid that dynamically resizes to any width.

Media Queries

Media queries allow defining different styles for different browser and device configurations.

Flexible Media

Images, videos, and other media types should also be dynamically resizable.

Some footer text.

Lab session 2: Responsive Web Design

Index

HTML5: SVG

Item 3

Item 4

Overview

Responsive web design makes your web page look good on all devices.

Responsive web design uses only HTML and CSS.

Responsive web design is not a program or a JavaScript.

Three principles

For a design to be called responsive, it has to adhere to three principles: **flexible layouts**, **media queries**, and **flexible media.**

Flexible layouts

Flexible layouts denote the practice of building the layout with a flexible grid, capable of dynamically resizing to any width.

Media Queries

Media queries provide the ability to specify different styles for individual browser and device circumstances, the width of the viewport or device orientation for example.

Flexible Media

Images, videos, and other media types need to be scalable, changing their size as the size of the viewport changes.

Flexible layouts

Layout should be a flexible grid that dynamically resizes to any width.

Media Queries

Media queries allow defining different styles for different browser and device configurations.

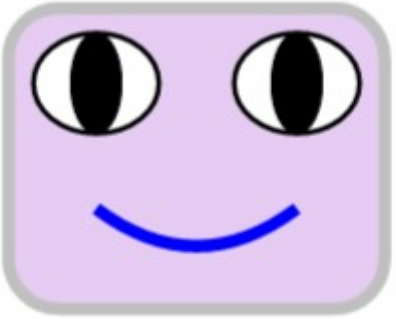
Flexible Media

Images, videos, and other media types should also be dynamically resizable.

Some footer text.

Assignment 2

- On page “HTML5: SVG”, use ellipse element to draw eyes on the smiley face
  - The end result should look something like the following



Assignment 3

- Implement assignment 1 using CSS3 **grid**
  - https://www.w3schools.com/css/css_grid.asp
  - https://medium.freecodecamp.org/learn-css-grid-in-5-minutes-f582e87b1228