POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

Bootstrap and Accessibility

Mattia Gianotti

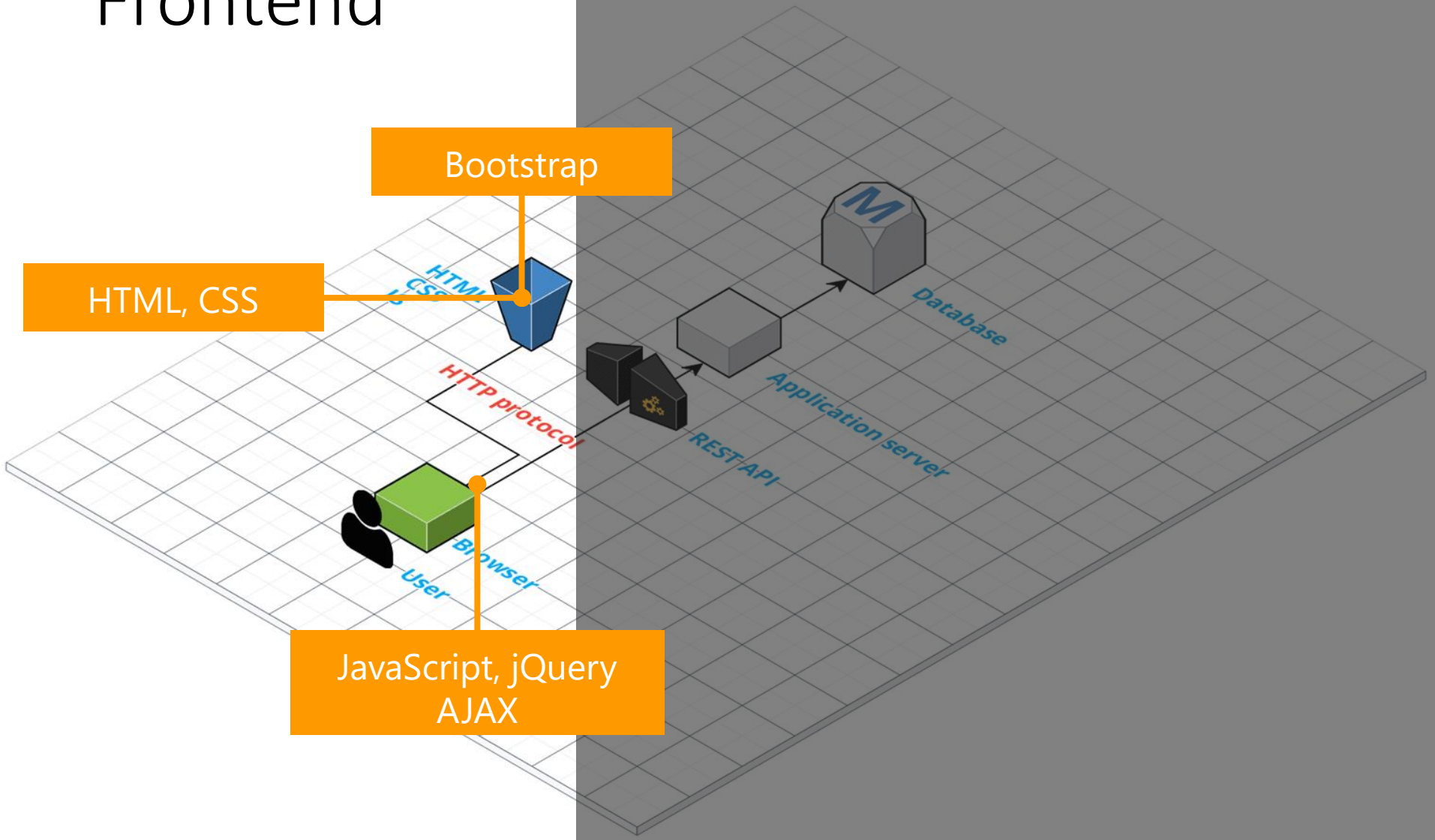
Hypermedia Applications (Web e Multimedia) - Prof. Franca Garzotto

2019-2020

Outline

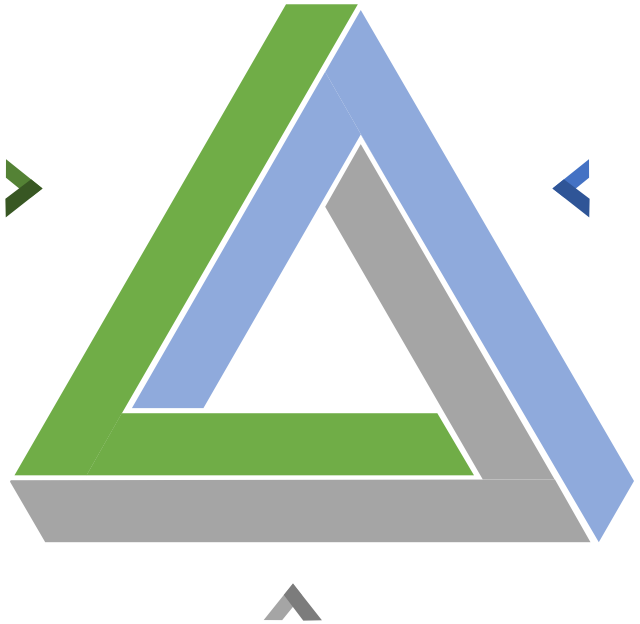
- Recap of CSS media query
- Bootstrap
  - Bootstrap basics and most used elements
  - Themes and Template
  - SaSS as an evolution of CSS
- Accessibility
  - Why accessibility is a problem
  - International standard for accessibility
  - Tools for accessibility verification

Frontend


Bootstrap
HTML, CSS
HTML
CSS
HTTP protocol
Database
Application server
REST API
Browser
User
JavaScript, jQuery
AJAX

JS

is a programming language that allows you to manipulate anything on the HTML page: the style, the contents of the page, but above all the interaction with the user. It allows us to create the logic of the UI and to take advantage of the API provided by the browser.



HTML

serves to define what elements are at stake, establish links (links) between the pages and the importance (not the shape or color) that the texts have, create forms for users, fix titles, upload images, videos, etc..

CSS

or "style sheets." It is a set of rules that allow you to define the look (style) that the elements on the page must take on. Dimensions, colors, animations, each visual characteristic can be manipulated.

Responsitivy in base CSS

- CSS can define a property when according to the media on which is shown. This is achieved using **media query**.

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

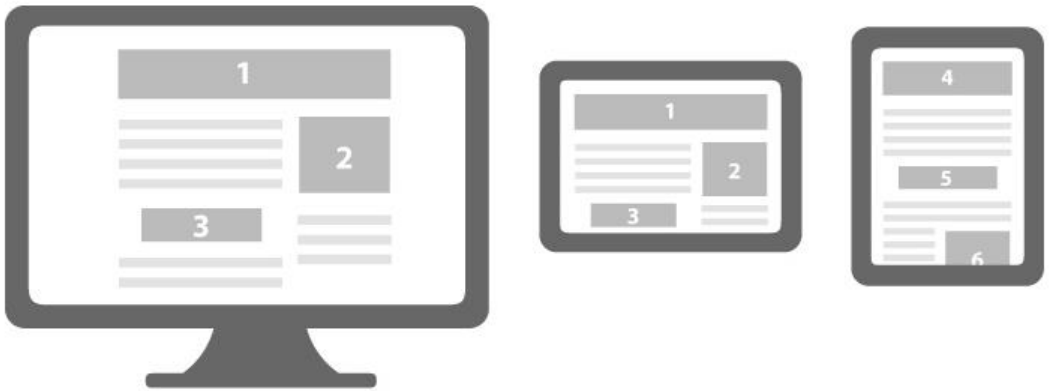
```
@media screen and (min-width: 480px) {
  #leftsidebar {width: 200px; float: left;}
  #main {margin-left: 216px;}
}
```

- It is even possible to load from links different CSS according to the media (RESPONSIVITY)

```
<link rel="stylesheet" media="mediatype and|not|only (expressions)" href="print.css">
```

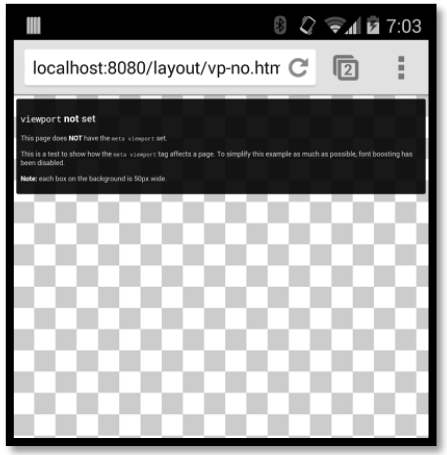
- For more reference:

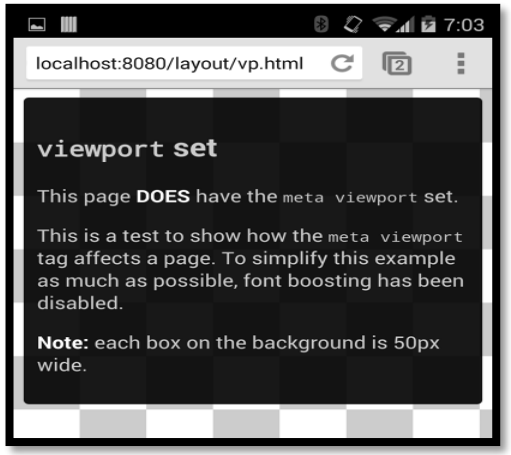
https://www.w3schools.com/css/css3_mediaqueries.asp

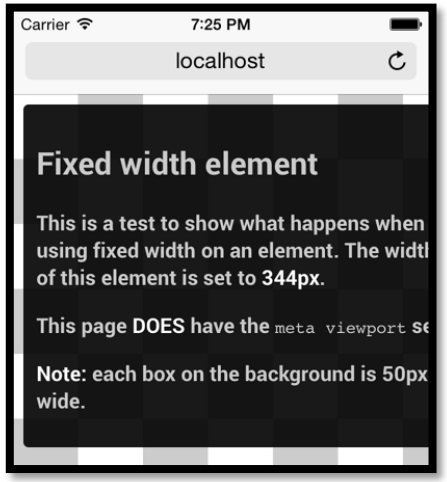


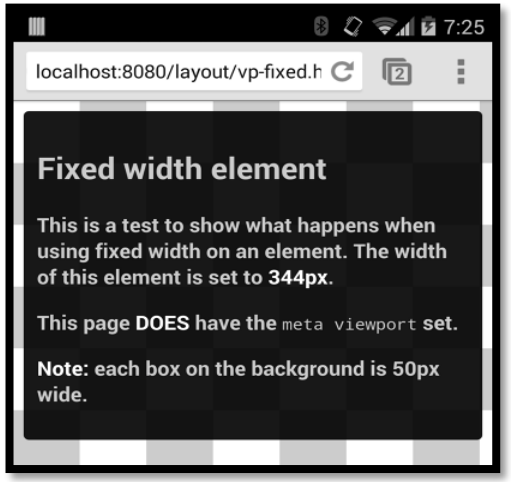
Tips for responsitivy in base CSS

- First initialize the viewport:
  - `<meta name="viewport" content="width=device-width, initial-scale=1">`
  - Use **meta viewport tag** to control the width and scaling of the browser's 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**.



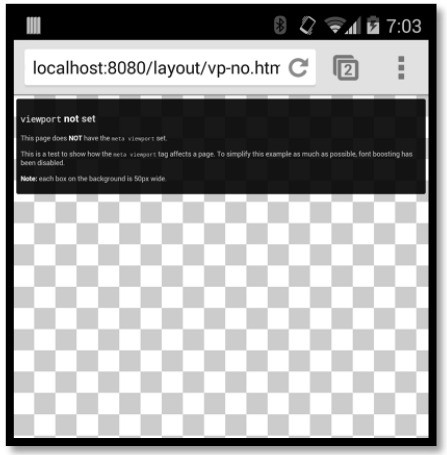


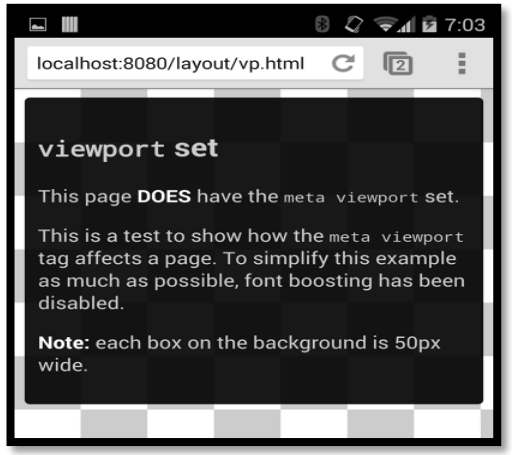


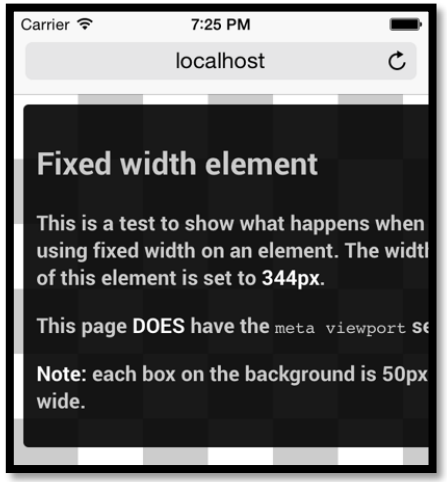


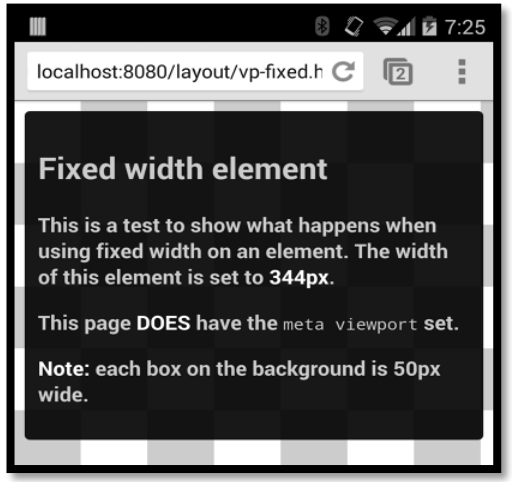
Tips for responsitivy in base CSS

- Secondly follow some tips:
  - **Do not use large fixed width elements**. And in general avoid fixed size. (use percentage or em units wherever possible)
  - 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.
  - Use standard measures https://css-tricks.com/snippets/css/media-queries-for-standard-devices/









Bootstrap

- Bootstrap is an **open source toolkit** for developing with HTML, CSS, and JS.
- Bootstrap is the most used **css framework** for responsive websites.
- Bootstrap is "**plug and play**" in your webpage.

- Bootstrap offers a lot of "prefabs" **components** you can use in your websites to speed up the process.
- Bootstrap offers you tons of examples and templates to start with.

- Alert
- Badge
- Breadcrumbs
- Button and Button groups
- **Card**
- **Carousel**
- Collapse
- **Dropdown**
- Form
- Iconography
- Input group
- Jumbotron
- List group
- Media object
- Modal
- Navs and **Navbar**
- Pagination
- Popovers
- Progress bar
- Scrollspy
- Spinners
- Toasts
- Tooltips

https://getbootstrap.com/docs/4.4/components/

Different alternatives to use bootstrap

- Bootstrap can be installed through npm to have access to the Sass and JS for your extreme customization (not very much suggested).
  - Documentation can be found at:
    https://getbootstrap.com/docs/4.4/getting-started/download/

- Bootstrap can be included in its compiled version.
  - Import the css.
    - <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
  - Add the proper remote JS needed by bootstrap.
    - <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymous"></script>
    - <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
    - <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>

- Start from a Bootstrap Template or Theme.

Pre-compiled Bootstrap

- Downloaded bootstrap has this **structure**.
- It is recommended to use the **minified version** of the bootstrap.min.cs and bootstrap.min.js at the production level for efficiency, but is less readable.
- For start you can use the **non-minified version**: same functions, more readable, just less efficient.
- Glyphicons **fonts** are included in some themes as optional.
- Bootstrap.*.map can be used by certain browser develop tools.

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

Bootstrap boilerplate code

```
<!doctype html>
<html lang="en">
 <head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
 <!—Insert your style sheets HERE -->

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

<!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>

<!– Insert your JavaScript HERE -->
 </body>
</html>
```

Usage of Bootstrap: Containers

- Containers are the most basic layout element in Bootstrap and are **required when using our default grid system**.
- Can be nested but in general is not useful.

- Bootstrap comes with three different containers (class):
  - **.container**, which sets a max-width at each responsive breakpoint (equal to .container-sm).
  - **.container-fluid**, which is width: 100% at all breakpoints.
  - **.container-{breakpoint}**, which is width: 100% until the specified breakpoint, where it assume a maximum size.
    - Breakpoint can be sm, md, lg, xl.

- Container div is the container of all responsive elements in your page.
- Official reference: https://getbootstrap.com/docs/4.4/layout/overview/

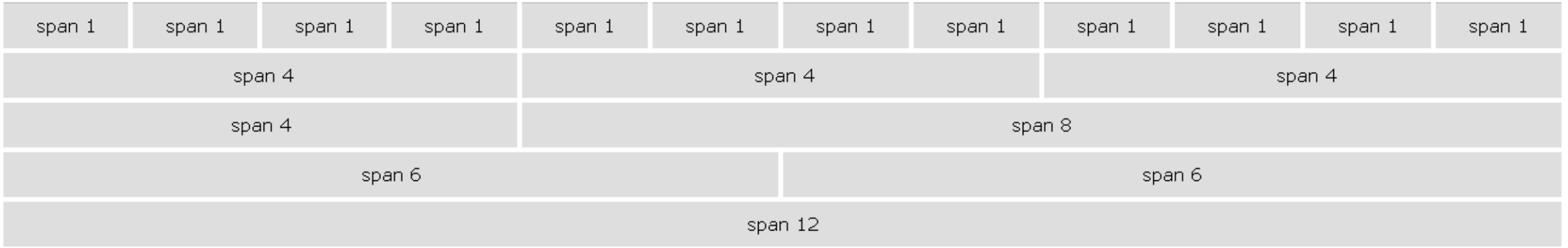
Bootstrap Grid System

Bootstrap's grid system uses a series of containers, rows, and columns to layout and align content. **It is fully responsive**.

The grid is divided in **rows**. Each row is then subdivided in **columns**.

Different number of columns can be added to each row.

Columns can be aligned with the class **align-self-X**, with X that can be end, start, center.
Rows can be aligned with the class **justify-content-Y**, with Y that can be start, center, end, around justify.



Basic Structure of a Bootstrap Grid

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

x can be **xs, sm, md** or **lg.**

***x* can be omitted** and use the auto extra small case.

y can be any value from 1 to 12 included. The sum of all ***y*** in a row **must be at most** 12.

Examples: 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>
```

Examples: Responsive queries

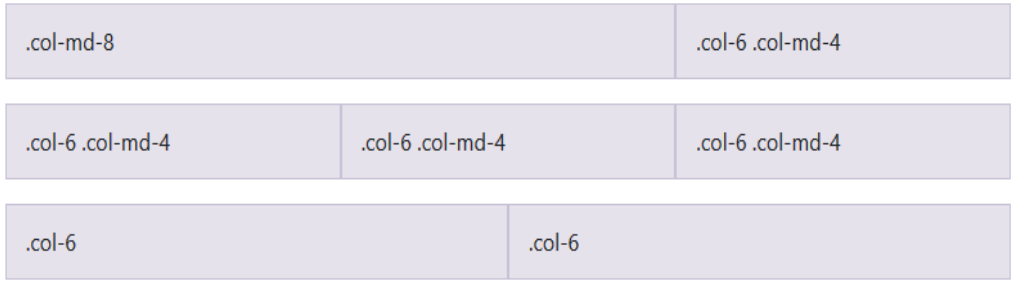
```
<div class="container">

        <div class ="row">
                <div class =«col-12 col-md-8">.col-md-8</div>
                <div class ="col-6 col-md-4">.col-6 .col-md-4</div>
         </div>


        <div class ="row">
                <div class ="col-6 col-md-4">.col-6 .col-md-4</div>
                <div class ="col-6 col-md-4">.col-6 .col-md-4</div>
                <div class ="col-6 col-md-4">.col-6 .col-md-4</div>
        </div>


         <div class ="row">
                <div class ="col-6">.col-6</div>
                <div class ="col-6">.col-6</div>
        </div>
</div>
```

Stack the columns on mobile by making one full-width and the other half-width

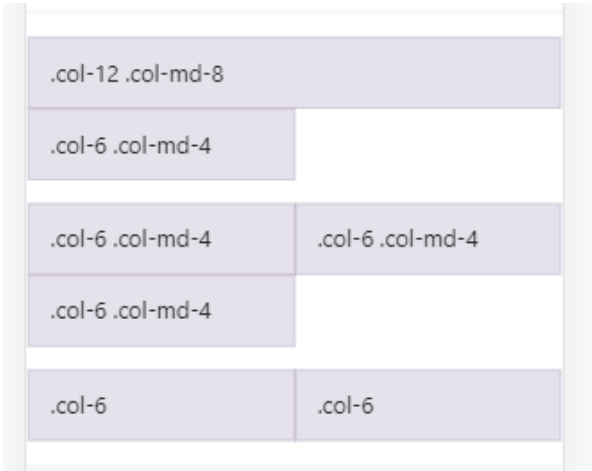
Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop

Columns are always 50% wide, on mobile and desktop

Desktop view

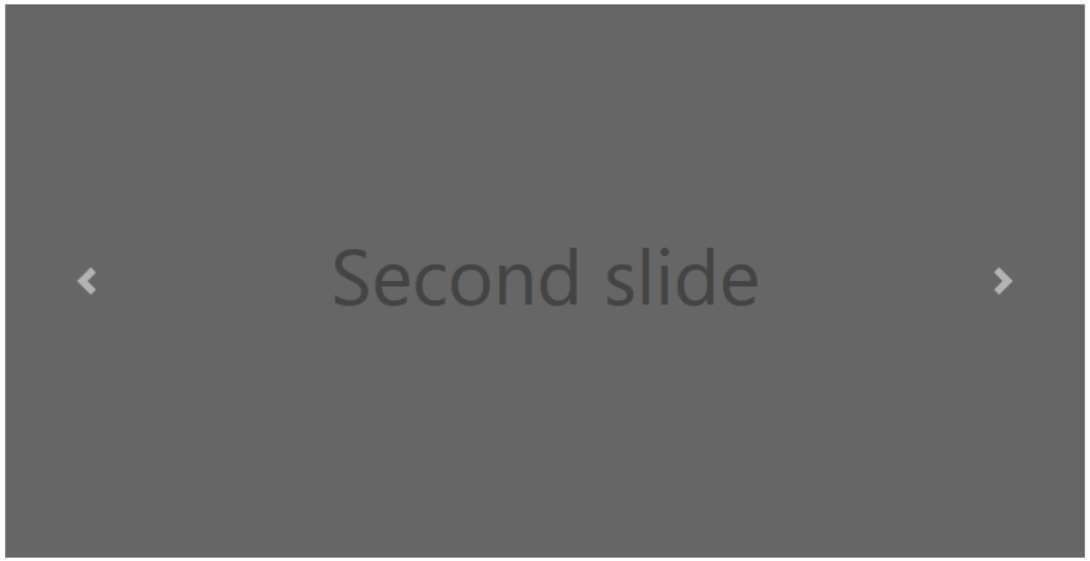


Phone view



Carousel

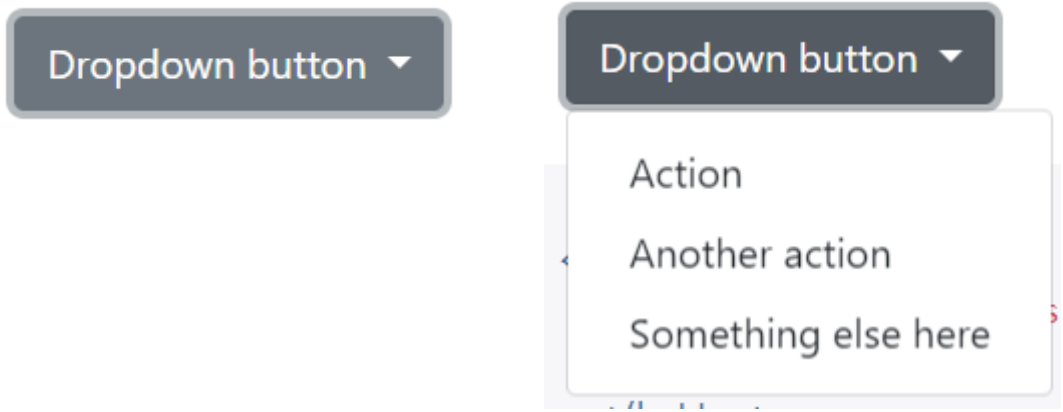
- The carousel is a slideshow for cycling through a series of content. It works with a series of images, text, or custom markup.
- It includes support for previous/next controls and indicators.




```
<div id="carouselExampleControls" class="carousel slide" data-ride="carousel">
        < div class="carousel-inner">
                < div class="carousel-item active">
                        <img src="..." class="d-block w-100" alt="...">
                </div>
                 < div class="carousel-item">
                        <img src="..." class="d-block w-100" alt="...">
                </div>
                < div class="carousel-item">
                        <img src="..." class="d-block w-100" alt="...">
                 </div>
         </div>
        <a class="carousel-control-prev" href="#carouselExampleControls" role="button" data-slide="prev">
                <span class="carousel-control-prev-icon" aria-hidden="true"></span>
                <span class="sr-only">Previous</span>
         </a>
        <a class="carousel-control-next" href="#carouselExampleControls" role="button" data-slide="next">
                <span class="carousel-control-next-icon" aria-hidden="true"></span>
                <span class="sr-only">Next</span>
         </a>
</div>
```

https://getbootstrap.com/docs/4.4/components/carousel/

Dropdown



- Dropdowns are toggleable, contextual overlays for **displaying lists of elements**, such as links, input,….
- They’re made interactive with the included Bootstrap **dropdown JavaScript plugin.**
- They’re **toggled** by clicking, not by hovering.
- Bootstrap’s dropdown **is not compliant** to Aria role = “menu” widget by default because it is more generic. Application of the standard accessible Aria labelling is left to the coders.
- However Bootstrap’s dropdown have built in support for **standard menu interaction** using keyboard (move among the items with cursor and exit with esc).

```
<div class="dropdown">
    <button class="btn btn-secondary dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
        Dropdown button
    </button>
    <div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
        <a class="dropdown-item" href="#">Action</a>
        <a class ="dropdown-item" href="#">Another action</a>
        <a class ="dropdown-item" href="#">Something else here</a>
    </div>
</div>
```

https://getbootstrap.com/docs/4.4/components/dropdowns/

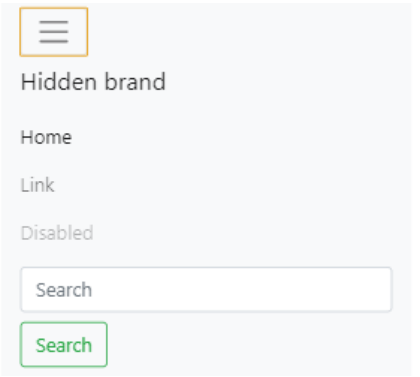
Navbar

- The navbar is the **navigation header**, used in almost all websites as the menu bar for the website.
- Navbars **require** a wrapping .navbar with .navbar-expand{-sm|-md|-lg|-xl} for responsive collapsing and color scheme classes.
- Use Bootstrap spacing and flex utility classes for **controlling spacing and alignment** within navbars.
- Ensure **accessibility** by using a <nav> element or, if using a more generic element such as a , add a role="navigation" to every navbar to explicitly identify it as a landmark region for users of assistive technologies.

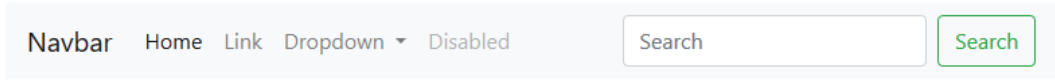
https://getbootstrap.com/docs/4.4/components/navbar/

Navbar example

Phone view



Desktop view

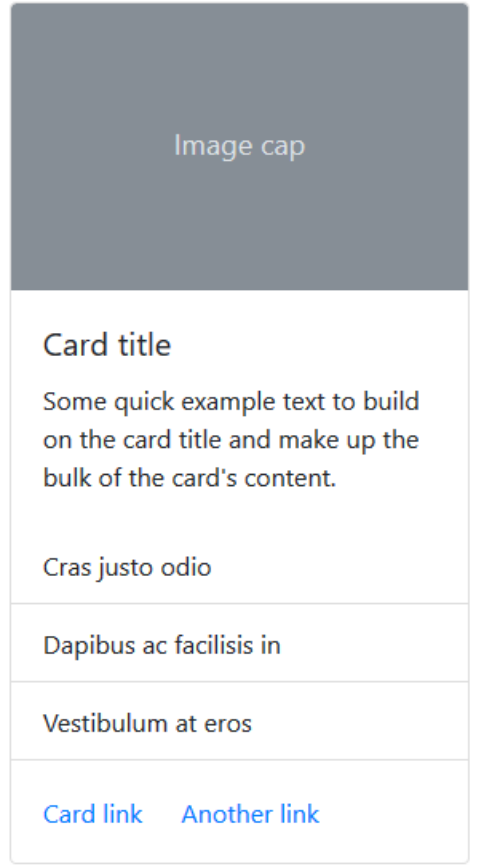


- Navbars come with built-in support for a handful of sub-components. Choose from the following as needed:
  - **.navbar-brand** used in general for logos and return to homepage.
  - **.navbar-nav** for a full-height and lightweight navigation (including support for dropdowns).
  - **.navbar-toggler** to be used for collapsible elements and other navigation toggling (e.g. collapsed menu on mobile view).
  - **.form-inline** for any form controls and actions.
  - **.navbar-text** for adding vertically centered strings of text.
  - **.collapse.navbar-collapse** for grouping and hiding navbar contents by a parent breakpoint.

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class ="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
        <span class ="navbar-toggler-icon"></span>
    </button>
    <div class ="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class ="navbar-nav mr-auto">
            <li class ="nav-item active">
                <a class ="nav-link" href="#">Home <span class ="sr-only">(current)</span></a>
            </li>
            <li class ="nav-item">
                <a class ="nav-link" href="#">Link</a>
            </li>
            <li class ="nav-item dropdown">
                <a class ="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-
haspopup="true" aria-expanded="false"> Dropdown </a>
                <div class ="dropdown-menu" aria-labelledby="navbarDropdown">
                    <a class ="dropdown-item" href="#">Action</a>
                    <a class ="dropdown-item" href="#">Another action</a>
                    <div class ="dropdown-divider"></div>
                    <a class ="dropdown-item" href="#">Something else here</a>
                </div>
            </li>
            <li class ="nav-item">
                <a class ="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Disabled</a>
            </li>
        </ul>
        <form class ="form-inline my-2 my-lg-0">
            <input class ="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search">
            <button class ="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
        </form>
    </div>
</nav>
```

Card

- Flexible and extensible **content container**.
- Offer **easy alignment** and mix well with other Bootstrap components.
- Card titles are used by adding **.card-title** to a <h*> tag.
- Links are added and placed next to each other by adding **.card-link** to an <a> tag.
- Subtitles are used by adding a **.card-subtitle** to a <h*> tag.
- Images can be added on top of the carded by using class **.card-img-top**
- With **.card-text**, text can be added to the card.
- Text within **.card-text** can also be styled with the standard HTML tags.




```
<div class="card">
    <img src="..." class ="card-img-top" alt="...">
    < div class ="card-body">
        <h5 class ="card-title">Card title</h5>
         <p class ="card-text">Some quick example text to build on the card title and make up
the bulk of the card's content.</p>
     </div>
    <ul class ="list-group list-group-flush">
        <li class ="list-group-item">Cras justo odio</li>
        <li class ="list-group-item">Dapibus ac facilisis in</li>
        <li class ="list-group-item">Vestibulum at eros</li>
    </ul>
    < div class ="card-body">
        <a href="#" class ="card-link">Card link</a>
        <a href="#" class ="card-link">Another link</a>
    </div>
</div>
```

https://getbootstrap.com/docs/4.4/components/card/

Bootstrap Themes and Templates

- A theme is a **fully designed** start-up site.
  - It may be very complete and full of availabilities.
  - Hardly is perfect fit for you.
  - Can be harder to modify.
  - You can find them here: https://startbootstrap.com/themes/
- A template is an **un-styled layout.**
  - It has nothing to suggest you which style you may desire to apply.
  - Is very easy to customize.
  - You can find them here: https://startbootstrap.com/templates/
- Both of them **contains** the required js and css linkage for Bootstrap.

SASS

- **S**yntactically **A**wesome **S**tyle**S**heet.
- Sass is an **extension** to CSS.
- Browsers **do not understand** Sass code. A Sass pre-processor is needed to **transpile** Sass code into standard CSS . (from terminal *sass input.scss output.css*).
- It is a useful tool to **manage and maintain** large CSS.
- Sass added **new features** to CSS.
- Sass files has the **".scss"** file extension.

https://www.w3schools.com/sass/

https://sass-lang.com/

Sass features

- **Variables**: it is possible to store values and reuse them in several rules. This approach allow for simpler distribution of the code.
  - Variables can be overwritten inside rules but has values only inside the rule itself.
  - The value !global after a value ignore the scope of the overridden function.

- **Nesting rule availability**: Saas allow to write nested rules of scss which will be transcribed applying the proper parenting modifies in css.
  - The option is available also for properties, so that it is possible to use the prefix as a property.
    - Example:

```
font: {
  family: Helvetica, sans-serif;
  size: 18px;
  weight: bold;
}
```

```
font-family: Helvetica, sans-serif;
font-size: 18px;
font-weight: bold;
```

- **Import**: allows to include different files into one other in order to decompose the style in packages and then pick up the ones you need. After compile a single file is realized, so a single http call is made.

- **Mixin**: define a set of properties to be included (with include directive) inside other classes.

- **Extend**: copy all the style of another directive and then allows to customize some elements.

How to import and customize a theme?

- **Create your Sass**. Then you can either:
  - **import** all bootstrap (not a good idea).
  - **import** only the modules you really need for your theme (suggested approach).
- **Customize** the map function and the variables as you want.
- **Override** the variables by giving new values before importing the block.
- **Add or remove** elements from predefined maps.
- **Add extra parts** to the CSS (such as custom classes) after the “/ Optional” section of the file.
- Remember to **recompile** the Sass file before using it.
- Official reference: https://getbootstrap.com/docs/4.4/getting-started/theming/

Bootstrap themes and templates

- https://wrapbootstrap.com/
- https://startbootstrap.com/
- https://bootswatch.com/
- Many more…

Accessibility

<< The power of the Web is in its universality.
Access by everyone regardless of disability is an essential aspect >>

Tim Berners-Lee, W3C Director and inventor of the World Wide Web

Why accessibility?

Web accessibility means that websites are **designed and developed** so that people **with disabilities** can use them. More specifically, people can:

- perceive, understand, navigate, and interact with the Web.
- contribute to the Web.

Disability comes in many **different forms:**

Visual

- Cataracts
- Sun glare
- Color blind
- Low vision
- Blind

Hearing

- Noise
- Ear infection
- Hard of hearing
- Deaf

Mobility

- Hands full
- Broken arm
- Spinal cord injury
- Amelia

Speech

- Ambient noise
- Speech impediment
- Unable to speak

Cognitive

- Sleepy
- Distraction
- Migraine
- Learning disabilities
- Autism
- Seizure

Neural

- Depression
- PTSD
- Bipolar
- Anxiety

AN ESTIMATED

ONE BILLION

PEOPLE IN THE WORLD HAVE A DISABILITY

39 million people worldwide are blind

Around 246 million people in the world have low vision

6 out of 10 people with learning disabilities need to wear glasses

10-15%

of people worldwide have dyslexia or learning difficulties

285 MILLION

people worldwide have a visual impairment

774 MILLION

people in the world cannot read or write

W3C Hosts the Web Accessibility Initiative (WAI)

The **Web Accessibility Initiative** (WAI):

- coordinates with all the other domains of the W3C.
- operates internationally in all three host sites of W3C.

WAI is **sponsored** by a variety of government and industry supporters of accessibility:

- Government:
  - U.S. Department of Education .
  - European Commission's Information Society Technologies Programme.
  - Canada's Assistive Devices Industry Office.
- Industry:
  - *CA*, *Fundación ONCE*, *HP*, *IBM*; *Microsoft Corporation*; *SAP*, *Verizon Foundation*, *Wells Fargo*

A Web Content Accessibility Guidelines (WCAG) has been developed. You can find reference at https://www.w3.org/TR/WCAG21/

P.O.U.R. principles

- **Perceivable:** Information and user interface components must be presentable to users in ways they can perceive.
- **Operable:** User interface components and navigation must be operable.
- **Understandable:** Information and operation of user interface must be understandable.
- **Robust:** Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technology.

Perciveable

- **Text Alternatives:** Provide **text alternatives** for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.
- **Adaptable:** Create content that can be presented in different ways (for example simpler layout) **without losing information** or structure.
- **Distinguishable:** Make it easier for users **to see and hear** content including separating foreground from background.
- **Use of Color:** Color is **not used as the only** visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.
- **Contrast (Minimum):** The visual presentation of text and images of text has a **contrast ratio** of at least 4.5:1
- **Resize text:** Except for captions and images of text, **text can be resized** without assistive technology up to 200 percent **without loss of content** or functionality.

Operable

- **Keyboard Accessible:** Make all functionality available **from a keyboard**.

- **Keyboard:** All functionality of the content is operable through a keyboard interface **without requiring specific timings** for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.

- **Enough Time:** Provide users **enough time** to read and use content.

- **Seizures and Physical Reactions:** Do not design content in a way **that is known** to cause seizures or physical reactions.

Understandable

- **Readable:** Make text content **readable and understandable**.
- **Predictable:** Make Web pages **appear and operate** in predictable ways.
- **Input Assistance:** Help users **avoid and correct** mistakes.

Robust

- **Compatible:** Content must be robust enough that it **can be interpreted reliably** by a wide variety of user agents, **including assistive technologies**.

- **Parsing:** In content implemented using markup languages:
  - elements have complete start and end tags.
  - elements are nested according to their specifications.
  - elements do not contain duplicate attributes, and any IDs are unique, except where the specifications allow these features.

- **Name, Role, Value:** For all user interface components (including but not limited to: form elements, links and components generated by scripts), the **name and role can be programmatically determined**; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies.

Aria labels

- WAI-ARIA, the **Accessible Rich Internet Applications** Suite, defines a way to make Web content and Web applications more accessible to people with disabilities by **defining how information** about the elements' functionality **can be provided to assistive technology**.

- Aria Labels are used to acquire Robust name, role and value.

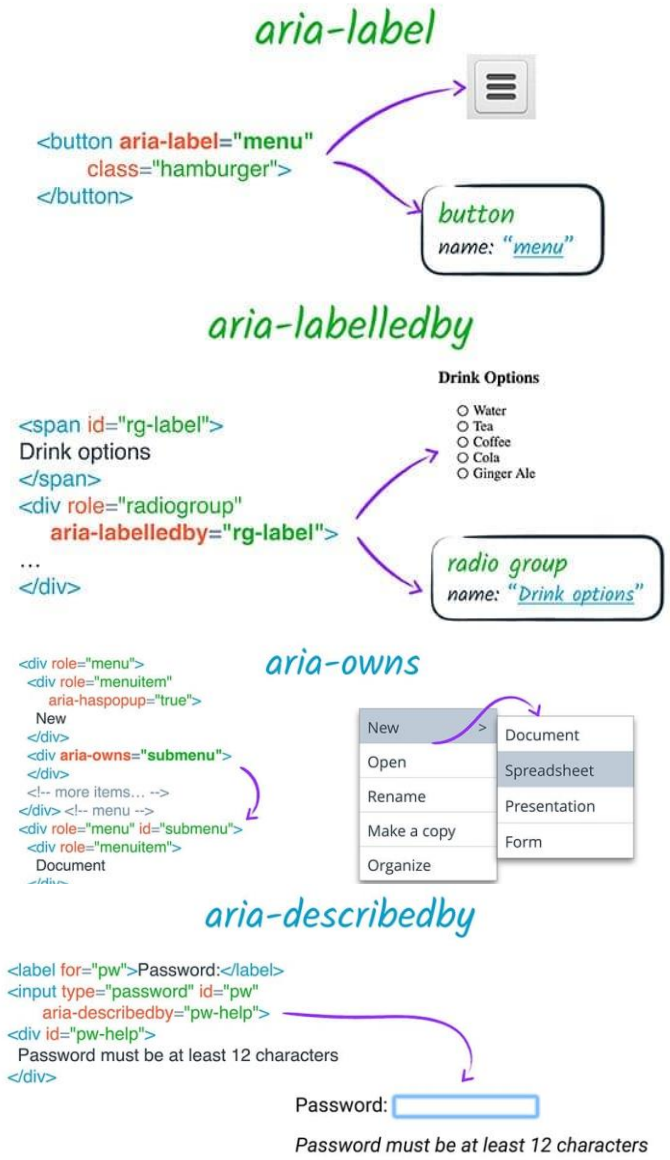
- Associate the aria-label attribute to a string which clearly describe the role of th tag you are using.

```
<div role="group" aria-labelledby="groupLabel">
                    <span id="groupLabel">Work Phone</span>
                    +<input type="number" aria-label="country code">
                    <input type=" number " aria-label="area code">
                    <input type=" number " aria-label="subscriber number">
</div>
```

- https://www.w3.org/TR/WCAG20-TECHS/ARIA14.html#top
- https://www.w3.org/WAI/WCAG20/quickref/20160105/#ensure-compat-rsv
- https://www.w3.org/TR/wai-aria-practices/#breadcrumb
- https://developers.google.com/web/fundamentals/accessibility/semantics-aria/aria-labels-and-relationships

Most common Aria Label

- **aria-label** allows us to specify a string to be used as the accessible label. This overrides any other native labeling mechanism.
- **aria-labelledby** allows us to specify the ID of another element in the DOM as an element's label.
- **aria-owns** allows us to tell assistive technology that an element that is separate in the DOM should be treated as a child of the current element, or to rearrange existing child elements into a different order.
- **aria-describedby** provides an accessible description in the same way that aria-labelledby provides a label.



Tools you can use

Check manually how does the website work with:

1. images turned off.
2. sound turned off.
3. larger than normal font sizes.
4. small screen resolution.
5. black and white display.
6. without a mouse.

Automatic testing f the website:

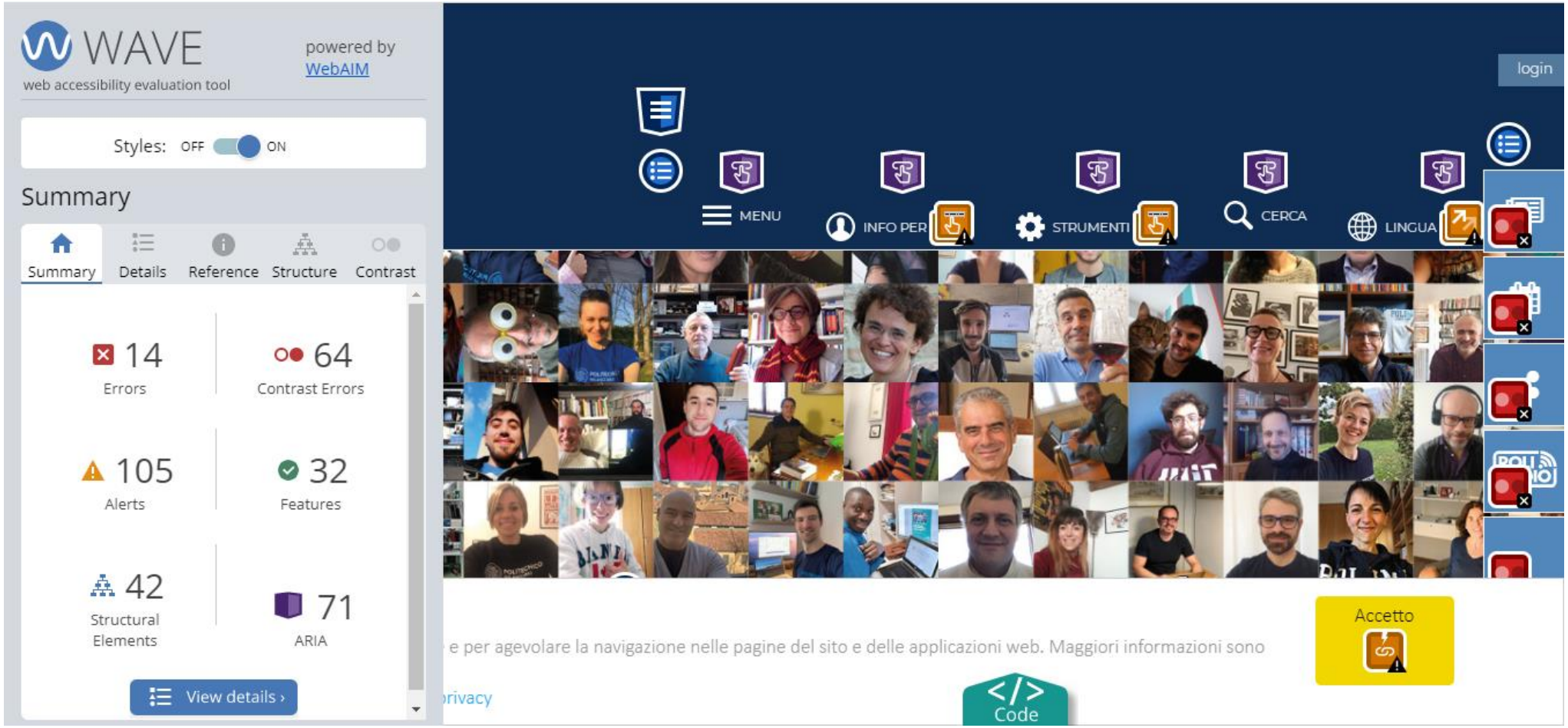
- Wave. (www.wave.webaim.org/)

Wave Tools usage

- Download and install Wave extension for Google Chrome.
- Open your website.

Wave Tools usage

- Open the extension on the page you want to analyse.



Wave Tools usage

- Check for the error, contrast error and alerts icon. If you ned you can also switch off the style of the webpage.
- In the detail panel by clicking on each symbol you can be sent to the webpage element presenting an error.
- In the contrast panel you can check what is the accessibility problem with the color contrast in your element.