Layout fundamentals for websites (NMW)

Lesson plan

Class 01

The basics of page layout

What is page layout?

- Comes from graphic design
- Refers to the arrangement and positioning of formal elements in a page (titles, text, images, etc.)
- Technical = In part instinctive (aesthetics)
  Artistic = Also supposes rules to be respected + techniques to be master
- Necess. qualities : Intelligence, sensibility, creativity,d sufficient culture, psychology,communication skills

Purpose of page layout

- Used to customize the appearance of various types of publications
- Gives a style, an ambiance + can add powerful connotations
- Has the power to give credibility to the brand, the company, the source of the communication
- Should take in consideration cognitive ergonomics :
  for the contents to have higher chances of being read + to reach its goals
- Important to consider different aspects before starting to design :
  Size and format of the viewports (desktop, mobile, tablet); static or dynamic?; adaptive or fluid;
  One-pager or multi-pages; resolution used for images, etc.

What does page layout include?

- First concerns = format of the medium (size and shape)
  The available space must be divided into thematic zones reserved for specific purposes.
- Second = positioning of the various elements of the composition (blocks of text, images, diagrams, captions, etc.), the negative space, the margins, the spacing and relations between the elements
- Usual principles of design must be applied, color scheme has to be chosen as well as well adapted fonts
- Must manage text contents = content hierarchy, levels of headings
  + text styles, text blocks, leading, kerning, lines length, etc.
  Layout grids should be produced and applied

Difference between printed and web page layout

- Printed document = static
  More control over the aesthetics, optimization of space (fixed) = graphic design and artistic skills

- Multimedia + web = dynamic => requires more various skills

  Example :
  websites elements need to be variable in size, scale and shape;
  may need to swap contents or for it's style to be personalized;
  some of them may use motion design, and needs to take navigation between contents in consideration, etc.

Phase of layout design

- Mandate definition
  - Produced with *a priori*, meeting with the client, research and documentation
  - A theoretical solution is elaborated (flow chart + different documents)
  - Solution is presented to the client for approval
- Ideation and conceptualization
  - Phase where ideas are generated to be used to later produce concepts
  - Techniques are use to generate initial ideas + moodboard
  - Ideas are put together and crossed to generate as many concepts as possible (rough sketches)
  - Selected concepts are evaluated = maximum 3 are finally selected => comprehensive mockups
  - Mockups usually include the approximate page layout
- Alpha and beta versions
  - Interface is segmented and integrated to have a working preliminary navigational version
  - Grids are developed and applied
  - Content is gradually integrated (layout is precisely determined)

Zen Garden project

400 different page layout concepts using the same (x)html. Only CSS file is different.
http://www.mezzoblue.com/zengarden/alldesigns/

Assignment 1 : Homepage design

Parameters to be defined.

Teacher gives the same subject to every student who must come up with their own unique home page designs.

Class 02

Popular page layout styles

Mondrian layout

- Arranged in a grid
- With or without rulers/gutters
- Type of like masonry

Picture window layout

- Big image
- Big title
- Little room for the body copy

Frame layout

- Images around the text
- May tell a story
- Can be BG image leaving space for text superimposed

Copy heavy layout

- Text takes most of the place
- Image can be used or not
- Break paragraphs, use sub-titles + subheads

Circus (or montage) layout

- Irregular composition of elements
- Some disorder slows down the reader = working through the disorder, the reader may remember more
- Filled with reverse blocks, oversize type, sunbursts, tilts, and assorted gimmicks
- Quite efficient in an ad for certain type of goods and customers

Multipanel layout layout

- Divides the composition into several surfaces (like comic strips)
  photos replacing the drawings, and text below the pictures replacing the balloons
- Panels are often of equal size (staccato effect keeping the reader moving effortlessly)
- Sometimes tell a story (can be used simply to display a series of products), pretty much in checkerboard fashion.

Silhouette layout

- Refers to the illustration/photo technique using the shadow or shape of an element
- Shape can be used to attract attention or as a container for other elements.

Big type layout

- Emphasizes on font style and sizes / main goal = grabbing attention
- Big type takes most of the space (leaving very little room for body copy)
  Body copy may be set in a type that is well beyond the normal 10- to 12-point)
- Type overpowers images here and images or illustration may not even be needed.
- Best big-type layouts use lowercase = because lower case is more interesting and easier to read
  When only few words are involved, it is possible to take some liberties with readability.

Rebus layout

- Images and text are forming a story
- Rebuses are small and simple images inserted at various places in text, sort of as visual puns.
- Image can be used to replace a word

Alphabet-inspired layout

- Alphabet inspired layout resembles to a story told
- Shape of letters may serve as a basic pattern in the positioning of the elements of a composition
- A composition designed to recreate the shape of a letter = strong unity and efficiently guides the reading.

Assignment 2: Home page layout style variations

Parameters to be defined.

Teacher gives the same subject to every student who must make 3 different mock-ups according to 3 different popular layout styles.

Class 03

Organizing content

- Homepage layout has more to do with image composition and titles positioning

- ALTHOUGH more and more websites offer contents right in the first page
  STILL COMMON for the home page to be mostly based on images
  + section pages to leave more space to contents of various nature

Full screen photo

- Spreading up content upon a full screen image

- Immediately anchors the subject of a site in the mind of a visitor + allows to tell a story

- Image is the most important element => text sections or menu sections are there to support it

- Sometimes don't have a menu (link to external pages like landing pages)

- Particularly efficient for websites focusing on a single type of content or a single product

Featured Image Layout

- One of the most frequent layouts

- Uses a representative featured image for each page of a website

- Image concentrates onto an expressive focal point
  Image = source of meaning.

100% horizontal

- Creates an elegant style using the entire page (often divided in different zones)
- This approach should be avoided for pages using vertical menus

Split screen layout (vertical or horizontal)

- Vertical split-screen generally conveys dual importance to two or more elements
  to favour quick choices (for higher engagement)

- Doesn't always really invite into making a choice, but sometimes to enhance one experience through the other

- Can help show two aspects of one thing while creating harmony by allowing to efficiently balance information

Asymmetrical Layout

- Sometimes slightly similar to split screen layout
- Needs for the designer to create active space, and to make white space livelier

Magazine Layout

- Imitates those of printed magazines
- Tends to be a mishmash of some other layouts
- All serving to give a splendour aura to the news.

Headline, thumbnails, grids and cards layout

- Information is organized into easy to browse grids
  = allowing equal distribution of text, photos, videos
  letting users decide upon the importance of each element
- Useful to present several types of contents or multiple products on the same level of importance
- Easy to manage in a responsive environment
- Headline and thumbnails gallery layouts = made up of image miniatures leading to full contents,
- + headlines (often along with short presentations) that serve as a guide though the image album

One-Column Layout

- Information is organized into one single column
- Content (text, photos, videos) is easy to follow
- Need to scan = identifying key points of interest into that single column.

The F Layout

- Based on psychometric studies + suitable for a wide range of websites
- The eyes move across a page in an F-shape pattern
- Often offers a detailed vertical menu on the left side (especially efficient for pages containing a lot of content)

Assignment 3 : Home page layout type variations

Parameters to be defined.

Teacher gives the same subject to every student who must make 3 different mock-ups according to 3 different popular layout types.

Class 04

Organizing content

- Web designers always have been using some kind of grids, managing to divide available space in different zones
- In the old days using frames and tables
- Now different CSS techniques = Semantic tags + containers

Modern grids

- Modern web grids = based on print grids used for centuries
  Explanation of basic grids system (Swiss school)
  Early grids = based on golden ratio

- Bringing the golden ratio to the web

- The formats (screens/viewports)
  Must be flexible (adaptive, fluid...)

- Using columns + gutters

- Using modules

- Web page analysis from a grid perspective

Creating a grid

- The format
  The columns
  12 / 16 ideally
  The margin and gutter

- The modular grid
  Modular grids possibilities

- Always sketch first

Assignment 4 : Finding the grid

Parameters to be defined.

Students chose a web page and trace the grid that was used for design

Class 05

Creating a CSS layout grid

12 columns layout grid

- Once sketch selected + Alpha version ready = integration of pages contents
- Use of a grid = assure cohesion + facilitate the developer's work

Step 1 : The grid container

- In a HTML5 page structure
- Create a container for the grid
- There are many different ways to create a layout grid = the easiest : grid display
- TO divide the pages width into 12 equal parts + 1rem gutter

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
</head>
<body>
<div class="grid">

</div>
</body>
</html>
```

CSS

```
.grid {
        display: grid;
        grid-template-columns: repeat(12, 1fr);
        grid-column-gap: 1rem;
        grid-row-gap: 1rem;
}
```

NOTA :
For a 960px (or 1200px, etc.) = add ***width: 100%*** + ***margin: auto***
Gutter is usually set to **20px**.

For a 16 columns grid = use ***16 instead of 12*** in the repeat function.

Step 2 : Spanning items horizontally (over columns/units)

- Add items into the grid container
- THEN span them over the wanted number of units (columns)

```
<div class="grid">
        <header> </header>
</div>
```

```
header {
        grid-column: span 12;
}
```

NOTA :

- No rows need to be created
- The sum of the units reach twelve
  (or sixteen, depending on the type of grid used)
- New row starts automatically.

Example :
For 3 items to use 4 units each, items need ***grid-column: span 4***
Maximum 12 units per row

HTML

```
<div class="grid">
    <div class="item"> </div>
    <div class="item"> </div>
    <div class="item"> </div>
</div>
```

CSS

```
.item {
        grid-column: span 4;
}
```

Spanning items vertically (over rows)

- To span vertically over rows = ***grid-row*** property must be used
  grid-row: span 3

HTML

```
<div class="grid">
    <div class="item"> </div>
    <div class="item2"> </div>
    <div class="item"> </div>
    <div class="item"> </div>
</div>
```

CSS

```
.item2 {
    grid-column: span 2;
    grid-row: span 3;
}
```

General advises

Responsive grids

- For a responsive grid = media-query

  Example :

```
@media only screen and (max-width: 768px) {
        .grid {
                display: flex;
                flex-direction: column;
        }
}
```

Margin

- Careful to enough gutter space = clear separations between the different zones
- Some containers need to have different margin adjustments made so texts align correctly (especially for titles).

Headings

- Such as <h1> = may have padding and margin = reset those to zero

Nested grids

- For a section (for instance the main section) = a nested grid can be used

Baseline grid

- Texts are shown side by side in two different container = use a baseline alignment

Assignment 5 : Creating and using a CSS layout grid

Parameters to be defined.

Students are given a web site page to redo using a CSS layout grid.

Class 06

Mid-term exam

Class 07

Bootstrap

- **What is Bootstrap :**
    - Front-end framework (HTML, CSS, JavaScript)
    - Free collection of tools to create websites + web applications
    - HTML + CSS based templates for various interface components
    - + optional JavaScript extensions
- **Advantages :**
    - Fast + efficient (time + money saving).
    - Responsive, mobile first, and tested
    - Everything you do and implement behave as expected.
    - No need to create complicated CSS (only needed to write the HTML)
    - Easy : based preexisting responsive CSS classes.
    - Large selection of website layouts, themes, admin panels, UI components, etc.
    - Only front-end framework supporting both LESS and SASS
- **Inconvenient :**
    - Many websites look very familiar, and dull (avoid using defaults)
    - Bootstrap is opinionated = opinion of what a website should be like, how layout should be managed... Difficult to do otherwise
    - Quite heavy : could be slow (make sure it is adapted to your target public)
    - Won't work if JavaScript is disabled and it doesn't provide fallbacks
      You have to provide your own CSS fallback if wanted

Quick start

- Download files =
  https://getbootstrap.com/docs/4.3/getting-started/download/

  OR
- CDN link =
  https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css

Basic Bootstrap implementation (CDN)

- HTML5 structure
- Paste necessary lines of codes in head section:

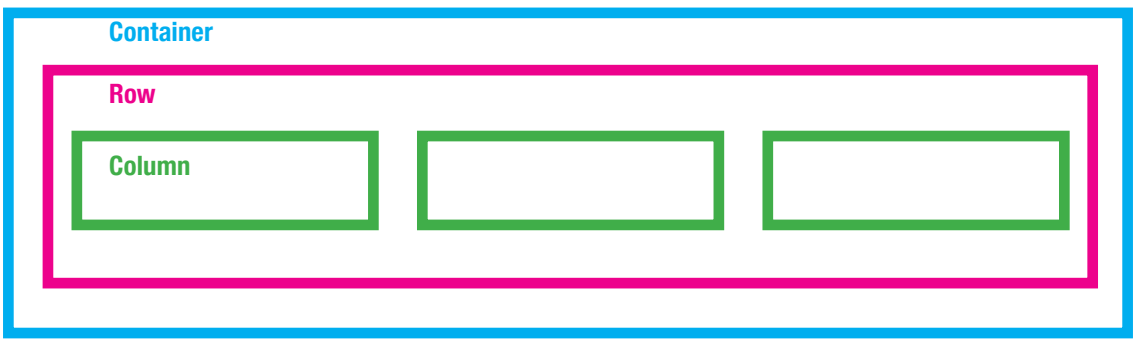
```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
```

Containers

- Two types of containers (both responsive): fixed or fluid
- Any tag can be used (ex.:)
- Fluid container => ***class="container-fluid"***
  (spans over the entire available width)
- Fixed width container => ***class="container"***
- Fixed container's width will vary depending on the size of the device.

Grid system

- Bootstrap includes a responsive, mobile first fluid grid system
  (scaling up to 12 columns as the device or viewport size increases)
- Offers predefined classes making page layout easy
- The container is divided in rows which are divided in columns
- To create a row => ***class="row"*** to the container



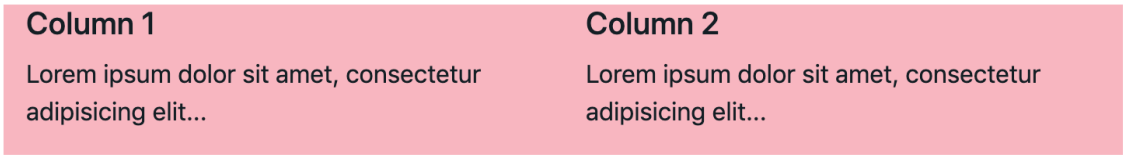
Grid's specificities

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
Container width	None (auto)	750px	970px	1170px
# of columns	12			
Column width	Auto	~62px	~81px	~97px
Gutter width	30px (15px on each side of a column)			
Nestable	Yes			
Offsets	Yes			
Column ordering	Yes			

Creating columns (units)

- To create column => ***class="col"*** to the items'
  (each of the items use an equal number of units with a 30px gutters - ***row-no-gutters*** for 0px gutter)

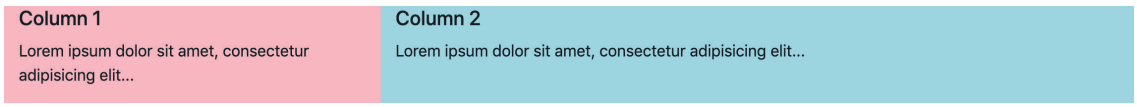
```
<div class="container">
        <div class="row">
                <div class="col">
                        <h5>Column 1</h5>
                        <p>Lorem ipsum dolor sit amet, consectetur adipis...</p>
                </div>
                <div class="col" >
                        <h5>Column 2</h5>
                        <p>Lorem ipsum dolor sit amet, consectetur adipis...</p>
                </div>
        </div>
</div>
```



Adjusting columns' width

- To adjust the width of an item => ***class="col-*"***
  (replace the star with the number of units (columns) the element should span over - Total of 12 units per row)

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



Responsive container measurements

- Media queries = usually used for a page to adapt to the size of the screen or viewport
- Bootstrap = viewports identified based on their sizes:

Extra small devices/xs (Phones)	< 768px
Small devices/sm (Tablets)	≥ 768px
Medium devices/md (Desktops)	≥ 992px
Large devices/lg (Desktops)	≥ 1200px

Responsiveness

- Classes are used to produce responsive layout

  Example:

  A responsive item

  Explanation:

  The items would occupy **two units** (columns) on *medium devices* (3 items per row)
  The items would occupy **twelve units** (one full-width item per row) on *small device*s

Column offset

- To move an items to the right in a row: class ***col-md-offset-****
  (replacing the star with a number representing how many units to moved away the item)

- ***col-md-offset-**** **THEN use** ***offset-md-2***

  Example:

```
<div class="container">
        <div class="row">
                <div class="col-md-2">My item 1</div>
                <div class="col-md-2 offset-md-2">My item 2</div>
        </div>
</div>
```

Nested columns

```
<div class="container">
        <div class="row">
                <div class="col-md-4">
                        <div class="row">
                                <div class="col-6">a</div>
                                <div class="col-6">b</div>
                        </div>
                </div>
        </div>
        <div class="col-md-8">My item</div>
</div>
```

Custom CSS

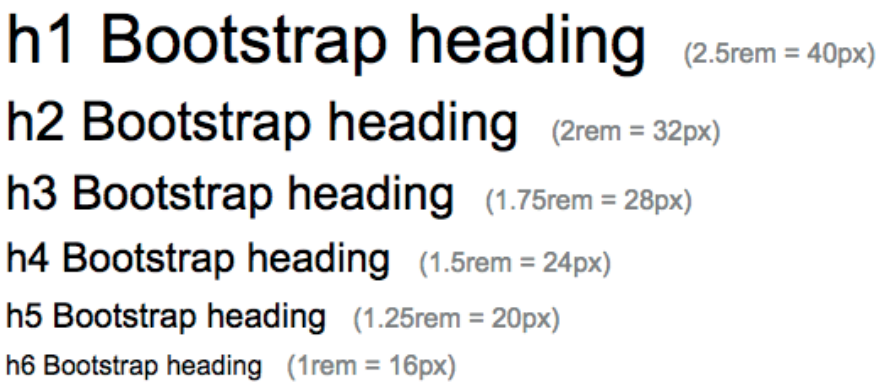
- To change existing Bootstrap CSS =
  link personal custom CSS after all the Bootstrap links in the <head> section

Text and typography

Generalities

Default font-size is set to *16px*.
Default line-height is set to *1.5rem*.
Default font-family is set to "*Helvetica Neue*", *Helvetica, Arial, sans-serif.*
All <p> elements are set with *margin-top: 0* and *margin-bottom: 1rem*.

Headings



Display headings

Display heading can be used when it is wanted for a heading to stand out more than usual headings (larger font-size and lighter font-weight).

```
<h1>Display Headings</h1>
<h1 class="display-1">Display 1</h1>
<h1 class="display-2">Display 2</h1>
<h1 class="display-3">Display 3</h1>
<h1 class="display-4">Display 4</h1>
```

Badges

Badges are used to supply supplementary information to any content in a contrasted shaded box.
<h1>Example heading <span **class="badge badge-secondary"**>New</h1>

Pill badge

<span **class="badge badge-pill badge-primary"**>Primary

Contextual classes (colors)

Text color :	
.text-muted	This text is muted
.text-primary	This text is primary
.text-success	This text is success
.text-info	This text is info
.text-warning	This text is warning
.text-danger	This text is danger
.text-secondary	This text is secondary
.text-white	This text is white
.text-dark	This text is dark
.text-body	(default body color/often black)
.text-light	This text is light

Background color

.bg-danger
.bg-info
.bg-primary
.bg-success

Text opacity

Black text with 50% opacity on white background
White text with 50% opacity on black background

Various tags and classes

<small>	Used to create a lighter, secondary text in any heading:
<mark>	Element with a pale yellow background color and some padding
<abbr>	Element with a dotted border bottom
.blockquote	Class to a <blockquote> for quoting blocks of content
.blockquote-footer	Class for the <footer> of a blockquote for author/source
.font-weight-bold	Class for bold text
.font-weight-bolder	Class for bolder text
.font-italic	Class for italic text
.font-weight-light	Class for light weight text
.font-weight-lighter	Class for lighter weight text
.font-weight-normal	Class for normal text
.lead	Class to make a paragraph stand out
.small	Class for smaller text (set to 80% of the size of the parent)
.text-left	Class for text to be left-aligned (center/right/justify)
.text-*-left	Class to align text on small, medium, large or xlarge (center/right/justify)
.text-break	Prevents long text from breaking layout
.text-decoration-none	Removes the underline from a link

.text-monospace	Class to make monospaced text
.text-nowrap	Class to apply no wrap text
.text-lowercase	Class to make lowercased text
.text-uppercase	Class to make uppercased text
.text-capitalize	Class to make capitalized text
.initialism	Class to make slightly smaller font size in a <abbr>
.list-unstyled	Removes the default list-style and left margin on list items Only applies to immediate children list items
.list-inline	Places all list items on a single line
.pre-scrollable	Makes a <pre> element scrollable

Borders

To add or remove predefined borders from a given container

```
<span class="border"></span>
<span class="border border-0"></span>
<span class="border border-top-0"></span>
<span class="border border-right-0"></span>
<span class="border border-bottom-0"></span>
<span class="border border-left-0"></span>
```

Border radius

```
<span class="rounded-sm"></span>
<span class="rounded"></span>
<span class="rounded-lg"></span>
<span class="rounded-top"></span>
<span class="rounded-right"></span>
<span class="rounded-bottom"></span>
<span class="rounded-left"></span>
<span class="rounded-circle"></span>
<span class="rounded-0"></span>
```

Assignment 6: Applying a Bootstrap layout grid

Parameters to be defined.

Students are given a web site page to build using a Bootstrap layout grid.

Images

Image Shapes

.rounded	Adds rounded corners to an image
.rounded-circle	Shapes the image to a circle
.img-thumbnail	Shapes the image to a thumbnail (bordered)

Aligning Images

Float an image to the right with the ***.float-right*** class or to the left with ***.float-left***.

Centered Image

Centers an image by adding the classes ***.mx-auto*** (margin:auto) and ***.d-block*** (display:block) to the image.

Responsive Images

For images to scale nicely to the parent element, the class **.img-fluid** can be used.

Class 08
Bootstrap (suite)

Button solid and outline styles

The button classes can be used on <a>, <button>, or <input> elements with contextual classes.

Solid buttons:

<button type="button" **class="btn"**>Basic</button>
<button type="button" class="**btn btn-primary**">Primary</button>

Outlined buttons:

<button type="button" class="**btn btn-outline-primary**">Primary</button>

Buttons sizes:

<button type="button" class="**btn btn-primary btn-lg**">Large</button>
<button type="button" class="**btn btn-primary**">Default</button>
<button type="button" class="**btn btn-primary btn-sm**">Small</button>

Notes:
btn-block creates a full width button
active class makes a button appear pressed
disabled class to make it visually appear disabled

Spinner buttons:

```
<button class="btn btn-primary">
        <span class="spinner-border spinner-border-md"></span>
</button>

<button class="btn btn-primary" disabled>
        <span class="spinner-border spinner-border-md"></span>
        Loading...
</button>

<button class="btn btn-danger">
        <span class="spinner-grow spinner-grow-md"></span>
        Loading...
</button>
```

Button groups :

```
<div class="btn-group-md">                                    * For vertical group : btn-group-vertical
        <button class="btn btn-primary">Apple</button>
        <button class="btn btn-primary">Samsung</button>
</div>

        Button group large :
        <div class="btn-group-lg">
                <button class="btn btn-primary">Apple</button>
                <button class="btn btn-primary">Samsung</button>
        </div>

        Button group small :
        <div class="btn-group-sm">
                <button class-sm="btn btn-primary">Apple</button>
                <button class="btn btn-primary">Samsung</button>
        </div>
```

Vertical button groups :

```
<div class="btn-group-vertical">
        <button type="button" class="btn btn-primary">Apple</button>
        <button type="button" class="btn btn-primary">Samsung</button>
</div>
```

Nesting Button Groups & Dropdown Menus

```
<div class="btn-group">
    <button type="button" class="btn btn-primary">Apple</button>
    <button type="button" class="btn btn-primary">Samsung</button>
    <div class="btn-group">
        <button type="button" class="btn btn-primary dropdown-toggle" data-toggle="dropdown">Sony</button>
        <div class="dropdown-menu">
            <a class="dropdown-item" href="#">Tablet</a>
            <a class="dropdown-item" href="#">Smartphone</a>
        </div>
    </div>
</div>
```

Split Button Dropdowns

```
<div class="btn-group">
    <button type="button" class="btn btn-primary">Sony</button>
    <button type="button" class="btn btn-primary dropdown-toggle dropdown-toggle-split" data-toggle="dropdown">
        <span class="caret"></span>
    </button>
    <div class="dropdown-menu">
        <a class="dropdown-item" href="#">Tablet</a>
        <a class="dropdown-item" href="#">Smartphone</a>
    </div>
</div>
```

Alert boxes

class="alert" + coloured using contextual classes

class="close" and **data-dismiss="alert"**

class=.alert-dismissible to the alert element for proper positioning of the close button

```
<div class="container">
    <div class="row">
        <div class="alert alert-success alert-dismissible">
            <button type="button" class="close"  data-dismiss="alert">&times;</button>
            <strong>Success!</strong> Indicates a successful or positive action.
        </div>
    </div>
</div>
```

Pagination

Basic Pagination

Add ***class="pagination"*** to a <ul> tag.
Then, add ***class="page-item"*** to each <li> tag
and ***class="page-link"*** to each link inside <li>.

```
<ul class="pagination">
        <li class="page-item"><a class="page-link" href="#">Previous</a></li>
        <li class="page-item active"><a class="page-link" href="#">1</a></li>
        <li class="page-item"><a class="page-link" href="#">2</a></li>
        <li class="page-item disabled"><a class="page-link" href="#">3</a></li>
```

```
        <li class="page-item"><a class="page-link" href="#">Next</a></li>
</ul>
```

Pagination sizing :

```
        <ul class="pagination pagination-lg">
        <ul class="pagination pagination-sm">
```

Pagination Alignment

Center-aligned :

```
<ul class="pagination justify-content-center" style="margin:20px 0">
        <li class="page-item">...</li>
</ul>
```

Right-aligned :

```
<ul class="pagination justify-content-end" style="margin:20px 0">
        <li class="page-item">...</li>
</ul>
```

Breadcrumbs

```
<ul class="breadcrumb">
        <li class="breadcrumb-item"><a href="#">United States</a></li>
        <li class="breadcrumb-item"><a href="#">Canada</a></li>
        <li class="breadcrumb-item"><a href="#">Germany</a></li>
        <li class="breadcrumb-item"><a href="#">Italy</a></li>
</ul>
```

Vertical list groups

Using <li> :

```
<ul class="list-group">
        <li class="list-group-item active">First item</li>
        <li class="list-group-item">Second item</li>
        <li class="list-group-item">Third item</li>
</ul>
```

Using <a> :

```
<div class="list-group">
```

```
        <a href="#" class="list-group-item list-group-item-action">Fist item</a>
        <a href="#" class="list-group-item list-group-item-action">Second item</a>
        <a href="#" class="list-group-item list-group-item-action">Third item</a>
</div>
```

Flush (remove borders):

```
<ul class="list-group list-group-flush">
```

Horizontal List Groups

```
<ul class="list-group list-group-horizontal">
        <li class="list-group-item">First item</li>
        <li class="list-group-item">Second item</li>
        <li class="list-group-item">Third item</li>
        <li class="list-group-item">Fourth item</li>
</ul>
```

Class 09

Bootstrap (suite)

Basic card

```
<div class="card">
    My card
</div>
```

Card with sections

```
<div class="card">
    <div class="card-header">Header</div>
    <div class="card-body">Body</div>
    <div class="card-footer">Footer</div>
</div>
```

Cards contents

```
<div class="card">
    <div class="card-header bg-warning">
        My card
    </div>
    <img class="card-img-top" src="bean.jpg" />
    <div class="card-body">
        <h4 class="card-title">Card title</h4>
        <p class="card-text">Your text here. And there...</p>
        <a href="#" class="card-link stretched-link">Website</a>
    </div>
    <div class="card-footer bg-info">
        Footer
    </div>
</div>
```

* ***stretched-link*** makes the entire card a link.
* ***card-img-overlay*** in the card-body section overlays the content over the image.