- Say Hello to HTML

Who am I?

Mariia Tarasiuk

Front-End developer
in EPAM

Key skill: FE + JS

Agenda

- Tags and doctype
- Semantics
- Accessibility
- Code convention
- Attributes
- Form
- Pictures
- Video/audio
- Meta
- Layout
- Box model
- Float vs flex vs grid
- Responsive

DEFINITIONS

- **HTML** - Hyper Text Markup Language
- **Tag** - basic element of html structure <tagname>content goes here...</tagname>
- **Attribute** - provides additional information about an element comes in name/value pairs like: name="value"

- `<p data-attr="some-value" class="class-attr"> Content</p>`

HTML basic structure

```
 <!DOCTYPE html>
<html>
    <head>
        <title>Page Title</title>
    </head>
    <body>

        <h1>My First Heading</h1>
        <p>My first paragraph.</p>

    </body>
</html>
```

Doctype

<!DOCTYPE [Top level element] [Publicity] "[Registration]//[Organization]//[Type] [Name]//[Lang]" "[URL]">

- A document type declaration, or DOCTYPE, is an instruction that associates a particular SGML or XML document (for example, a webpage) with a document type definition (DTD) (for example, the formal definition of a particular version of HTML)

- **Top level element** — It indicates the top-level element in a document to HTML, this tag <html>
- **Publicity** — PUBLIC or SYSTEM
- **Registration** — Developer DTD registered with the International Organization for Standardization. It takes one of two values: plus - a developer registered in the ISO, and minus - the developer is not registered. The W3C value put "-".
- **Organization** — unique name of the organization that developed the DTD
- **Type** — type describes the document. For HTML value specified DTD.
- **Name** — unique name to describe the document DTD.
- **Lang** — the language in which the text is written to describe the object
- **URL** — address of the document with DTD.

Types of tags

Метаданные документа	<html>,<head>,<title>,<base>,<link>,<meta>,<style>
Секционные элементы	<body>,<article>,<section>,<nav>,<aside>,<header>,<footer>
Группировка содержимого	<div>,<p>,<address>,<blockquote>,<figure>, all list tags
Табличные данные	<caption>, <table> and table related tags
Интерактивные элементы	<details>,<summary>,<dialog>
Скрипты	<script>,<template>,<canvas>
Встроенное содержимое	<picture>,<source>,<iframe>,<embed>,<object>,<audio>,<video>
Семантика текста	<cite>,<abbr>,<data>,<time>,<a>
Формы	<form> and form related tags
Правки в тексте	<del>,<ins>

HTML5 Semantic Elements

Semantic HTML elements clearly describe it’s meaning in a human and machine readable way

It is much **easier to read** **and** has **greater accessibility**

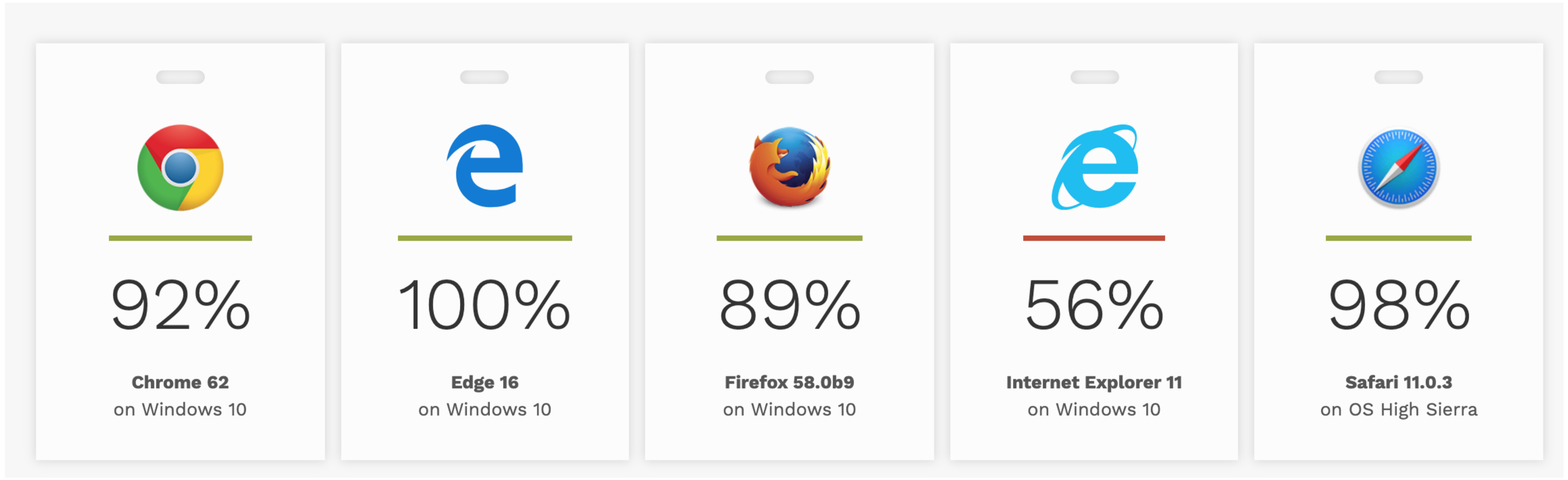
Compare

```
<header></header>
<section>
    <article>
        <figure>
            <img>
            <figcaption></figcaption>
        </figure>
    </article>
</section>
<footer></footer>
```

```
<div id="header"></div>
<div class="section">
    <div class="article">
        <div class="figure">
            <img>
            <div class="figcaption"></div>
        </div>
    </div>
</div>
<div id="footer"></div>
```

HTML accessibility

- Semantic HTML
  - <button>Click Me</button> vs ~~<div>Click Me</div>~~
- Use Headings
- Alternative Text for images
- Declare the Language
  - <html lang="en">
- Use clear language that is easy to understand, and try to avoid characters that cannot be read clearly by a screen reader.
  - Keep sentences as short as possible.
  - Avoid dashes. Instead of writing 1-3, write 1 to 3
  - Avoid abbreviations. Instead of writing Feb, write February
  - Avoid slang words
- A link should explain clearly what information the reader will get by clicking on that link
  - Find out more about the HTML language vs ~~Click here~~
- Link Titles. The title attribute specifies extra information about an element
- Using native keyboard accessibility
- WAI-ARIA (Web Accessibility Initiative - Accessible Rich Internet Applications)
- Reserved characters in HTML must be replaced with character entities



92%	100%	89%	56%	98%
Chrome 62 on Windows 10	**Edge 16** on Windows 10	**Firefox 58.0b9** on Windows 10	**Internet Explorer 11** on Windows 10	**Safari 11.0.3** on OS High Sierra

Current accessibility support status of HTML5 features across major browsers

W3C
Web Standarts
(W3C, WHATWG)

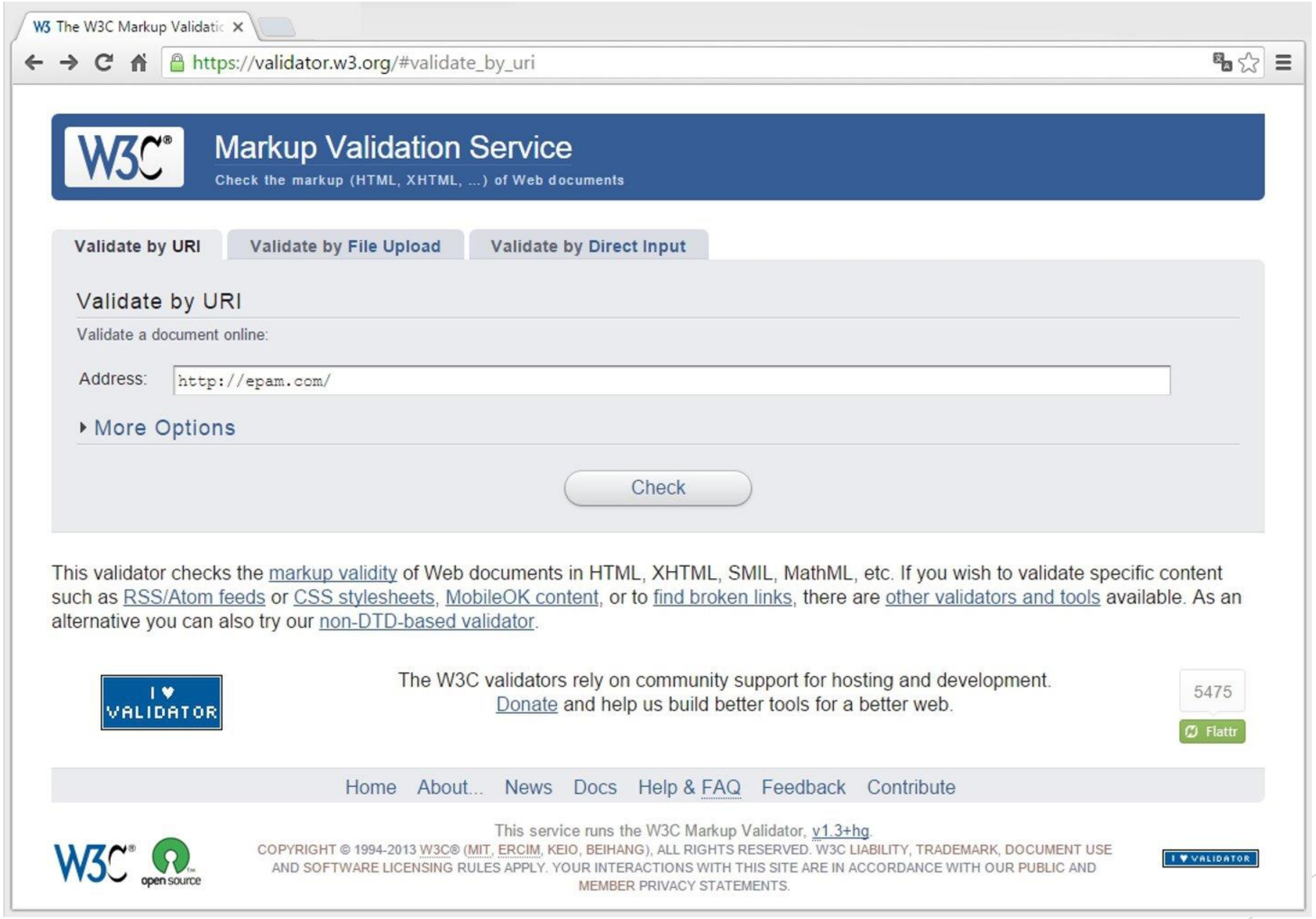
Company
Company Code Conventions

Project
Project Code Conventions

Team
Team Code Conventions

HTML Coding Conventions

VALIDATION



Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI

Validate a document online:

Address: http://epam.com/

This validator checks the markup validity of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as RSS/Atom feeds or CSS stylesheets, MobileOK content, or to find broken links, there are other validators and tools available. As an alternative you can also try our non-DTD-based validator.

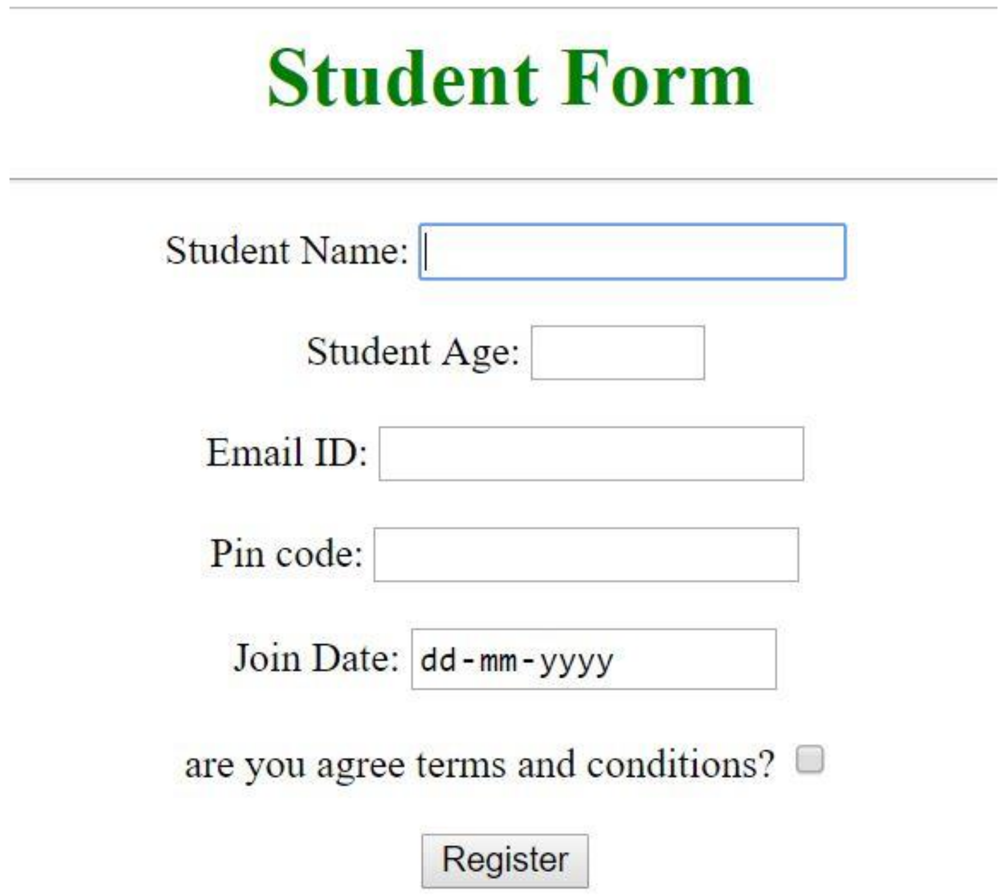
The W3C validators rely on community support for hosting and development.
Donate and help us build better tools for a better web.

This service runs the W3C Markup Validator, v1.3+hg.

COPYRIGHT © 1994-2013 W3C® (MIT, ERCIM, KEIO, BEIHANG), ALL RIGHTS RESERVED. W3C LIABILITY, TRADEMARK, DOCUMENT USE AND SOFTWARE LICENSING RULES APPLY. YOUR INTERACTIONS WITH THIS SITE ARE IN ACCORDANCE WITH OUR PUBLIC AND MEMBER PRIVACY STATEMENTS.

Form

- a group of user input (UI) controls that a ccepts information from the user and sends the information to a web server


Student Form

Student Name:

Student Age:

Email ID:

Pin code:

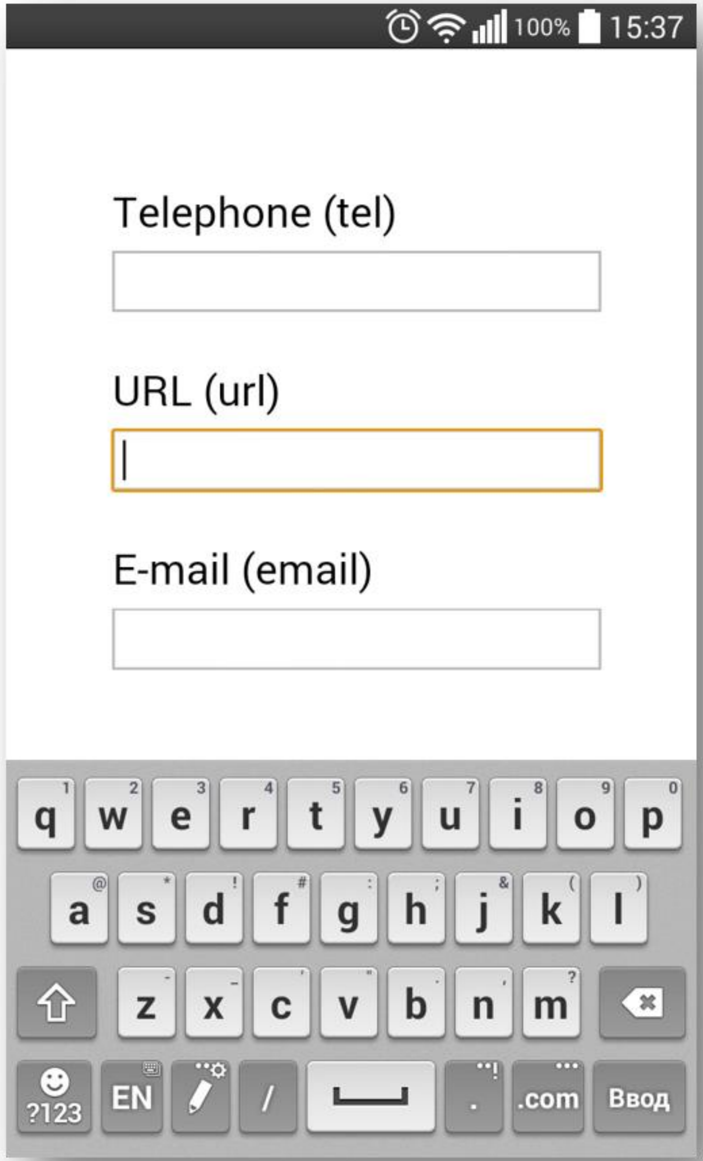
Join Date: dd-mm-yyyy

are you agree terms and conditions?

- HTML has tags to create a collection of objects that implement this information gathering
- These objects are called ***widgets*** or ***controls*** or ***components*** (e.g., **buttons**, **checkboxes**, **text fields**, etc.)
- When the *Submit* button of a form is clicked, the form's values are sent to the server for processing
- The **`method`** attribute of <**`form`**> specifies one of the two possible techniques of transferring the form data to the server, **`get`** and **`post`**
- The information is sent to the server as a **query string** parameter embedded in the URL of the HTTP get or in the body of the HTTP post request.
- If the form has no action, the value of **`action`** is the empty string
- One page may contain many forms if so desired

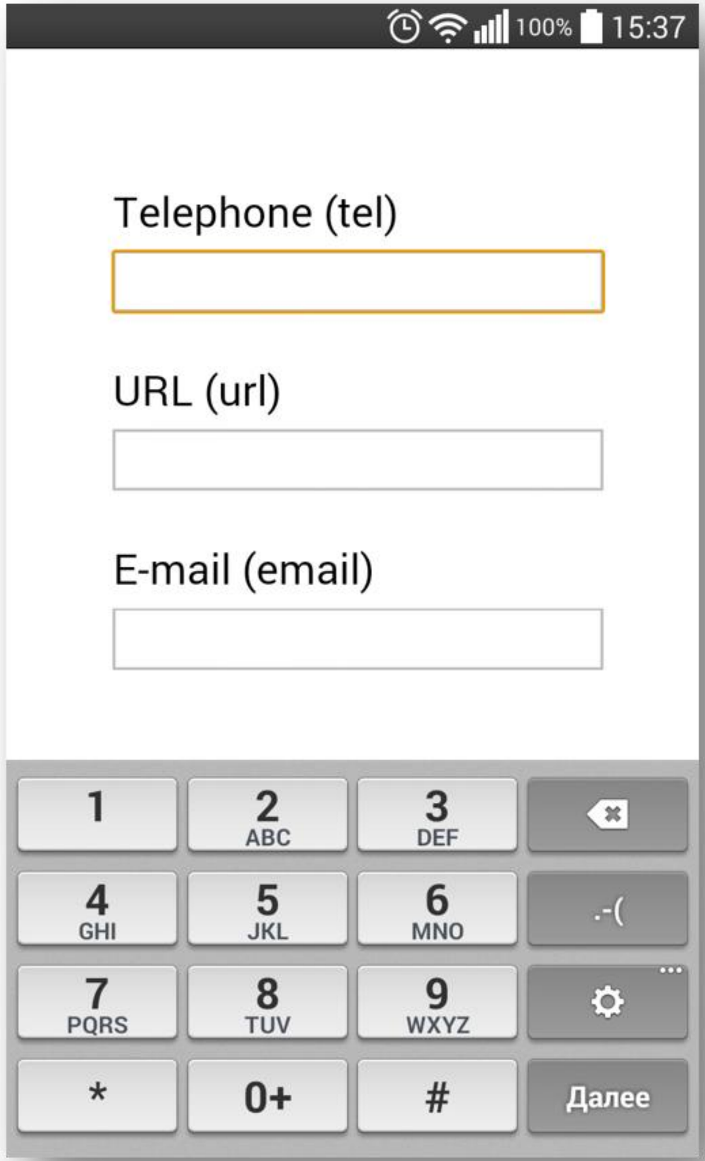
Form controls: <input>

- button
- checkbox
- *color (HTML5)*
- date (HTML5)
- datetime-local (HTML5)
- email (HTML5)
- file
- hidden
- image
- month (HTML5)
- number (HTML5)
- password
- radio
- range (HTML5)
- reset
- search (HTML5)
- submit
- tel (HTML5)
- text
- time (HTML5)
- url (HTML5)
- week (HTML5)


Telephone (tel)

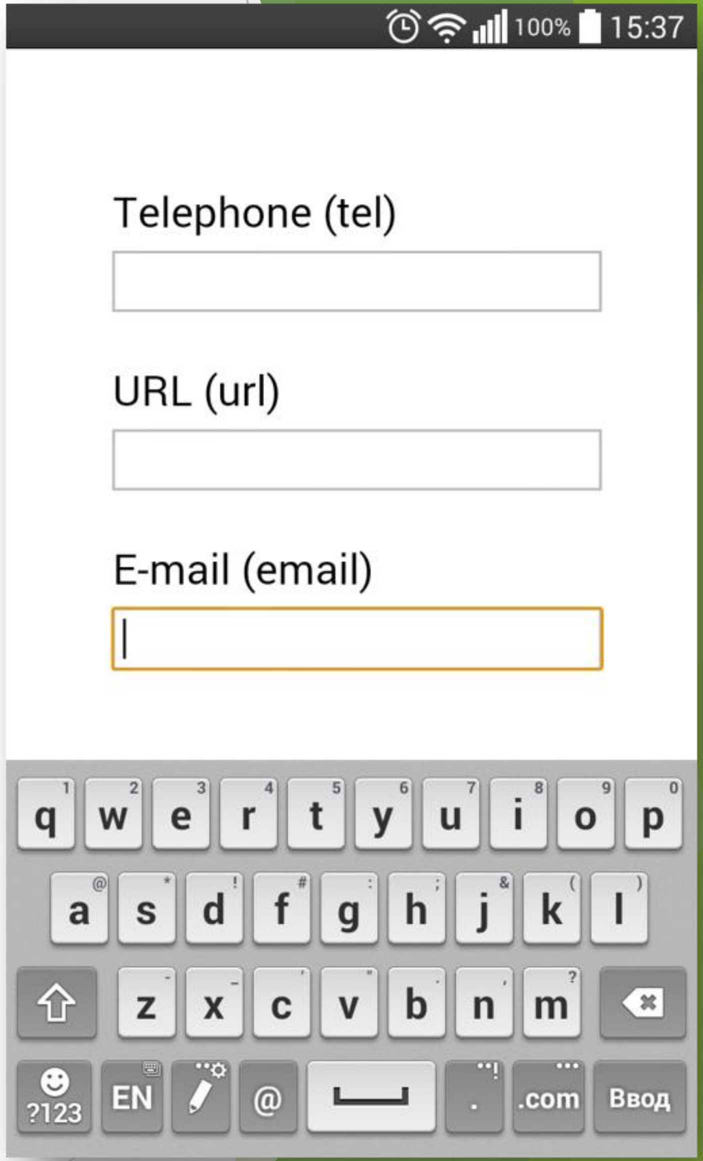
URL (url)

E-mail (email)


Telephone (tel)

URL (url)

E-mail (email)


Telephone (tel)

URL (url)

E-mail (email)

The autocomplete Attribute

- The autocomplete attribute specifies whether a form or input field should have autocomplete on or off.
- When autocomplete is on, the browser automatically completes the input values based on values that the user has entered before.
- **Tip:** It is possible to have autocomplete "on" for the form, and "off" for specific input fields, or vice versa.
- The autocomplete attribute works with <form> and the following <input> types: text, search, url, tel, email, password, datepickers, range, and color

```
<form action="/action_page.php" autocomplete="on">
  First name:<input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  E-mail: <input type="email" name="email" autocomplete="off"><br>
  <input type="submit">
</form>
```

The autofocus Attribute

- <input type="text" name="fname" autofocus>
- The autofocus attribute specifies that the input field should automatically get focus when the page loads

The min and max Attributes

- Enter a date before 1980-01-01:
  <input type="date" name="bday" max="1979-12-31">

  Enter a date after 2000-01-01:
  <input type="date" name="bday" min="2000-01-02">

  Quantity (between 1 and 5):
  <input type="number" name="quantity" min="1" max="5">

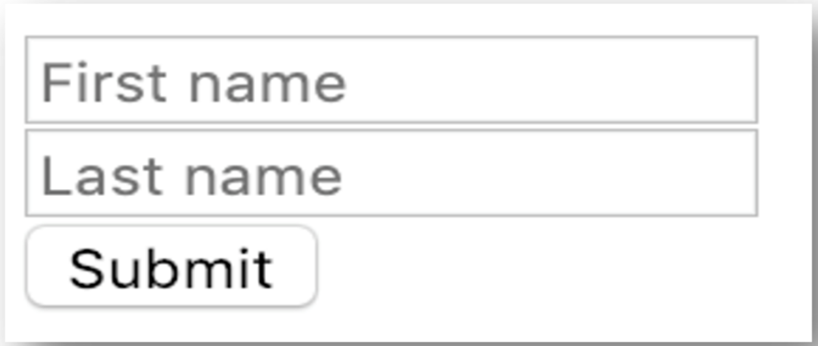
- The min and max attributes specify the minimum and maximum values for an <input> element.
- The min and max attributes work with the following input types: number, range, date, datetime-local, month, time and week.

The pattern Attribute

- The pattern attribute specifies a regular expression that the <input> element's value is checked against.
- The pattern attribute works with the following input types: text, search, url, tel, email, and password.
- An input field that can contain only three letters (no numbers or special characters):
- <input type="text" name="country_code"
 pattern="[A-Za-z]{3}" title="Three letter country code">

The placeholder Attribute

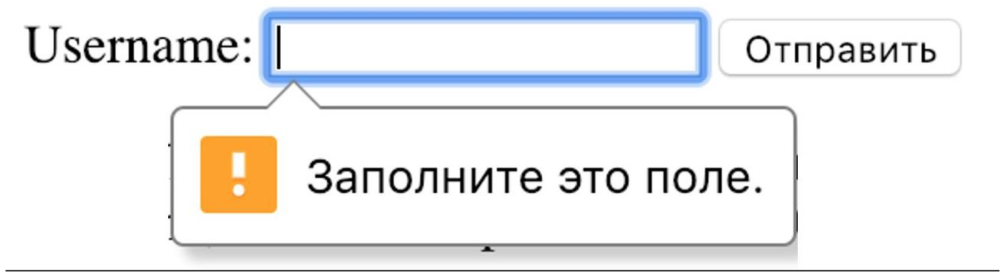
- The placeholder attribute specifies a hint that describes the expected value of an input field (a sample value or a short description of the format).
- The hint is displayed in the input field before the user enters a value.
- `<input type="text" name="fname" placeholder="First name">`
- `<input type="text" name="lname" placeholder="Last name">`



The required Attribute

- The required attribute specifies that an input field must be filled out before submitting the form.
- The required attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.

```
<input type="text" name="username" required>
```

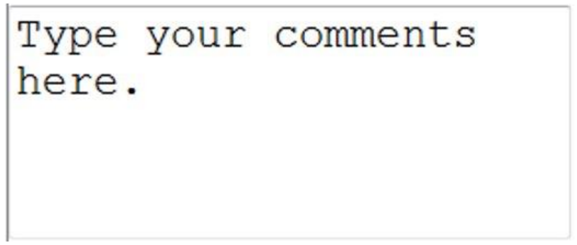


Заполните это поле.

Text boxes: <textarea>

- a multi-line text input area (inline)
- initial text is placed inside **`textarea`** tag (optional)
- required **`rows`** and **`cols`** attributes specify height/width in characters
- optional **`readonly`** attribute means text cannot be modified
- by default the text wrap to the next line.

```
<textarea rows="4" cols="20">
    Type your comments here.
</textarea>
```

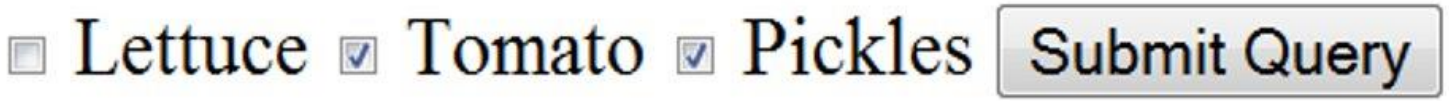


```
Type your comments
here.
```

Checkboxes: <input>

- yes/no choices that can be checked and unchecked (inline)
- none, 1, or many checkboxes can be checked at same time
- when sent to server, any checked boxes will be sent with value **on**:
  - http://www.cs.aub.edu.lb/params.php?tomato=on&pickles=on
- use **checked="checked"** attribute in HTML to initially check the box

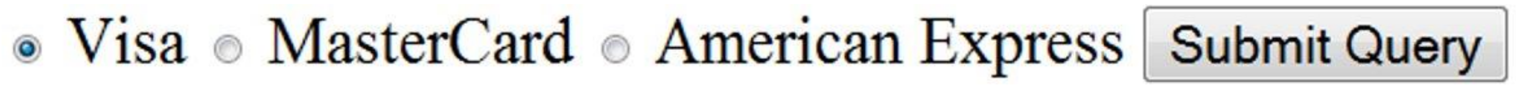
```
<input type="checkbox" name="lettuce" /> Lettuce
<input type="checkbox" name="tomato" checked="checked" /> Tomato
<input type="checkbox" name="pickles" checked="checked" /> Pickles
<input type="submit" />
```



Radio buttons: <input>

- sets of mutually exclusive choices (inline)
- grouped by name attribute (only one can be checked at a time)
  - So the browser will ensure that only one radio buttons can be checked from a group
- **must specify a value for each one** or else it will be sent as value **on**

```
<input type="radio" name="cc" value="visa" checked="checked" /> Visa
<input type="radio" name="cc" value="mastercard" /> MasterCard
<input type="radio" name="cc" value="amex" /> American Express
<input type="submit" />
```



Text labels: <label>

- associates nearby text with control, so you can click text to activate control
- can be used with checkboxes or radio buttons
- label element can be targeted by CSS style rules
- It can nest a control element
- Also you can **give each a control an** id and then a label placed anywhere in the page can target that control by placing a `for` attribute inside it and giving that attribute the same value as the id.

```
<label> <input type="radio" name="cc" value="visa"
checked="checked"/> Visa</label>
```

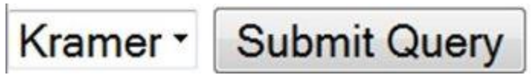
OR

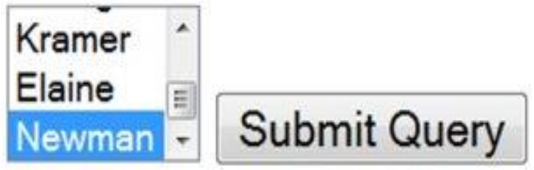
```
<input id="boldness"  type="checkbox" name="bold"  /> Visa
…
<label for="boldness"> click me to check the bold box </label>
```

Drop-down list: <select>, <option>

- menus of choices that collapse and expand (inline)
- `option` element represents each choice
- `select` optional attributes: `disabled`, `multiple`, `size`
- optional `selected` attribute sets which one is initially chosen
- optional `multiple` attribute allows selecting multiple items with **shift- or ctrl-click**
  - must declare parameter's name with `[]` if you allow multiple selections
- option tags can be set to be initially `selected`

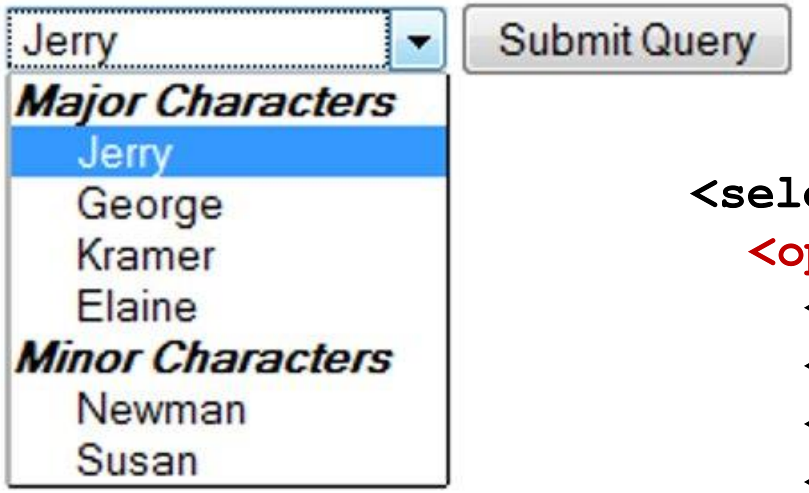
```
<select name="favoritecharacter">
  <option>Jerry</option>
  <option>George</option>
 <option selected="selected">Kramer</option>
  <option>Elaine</option>
</select>
<input type="submit" />
```





```
<select name="favoritecharacter[]" size="3" multiple="multiple">
  <option>Jerry</option>
  <option>George</option>
  <option>Kramer</option>
  <option>Elaine</option>
  <option selected="selected">Newman</option>
</select>
<input type="submit" />
```

Option groups: <optgroup>



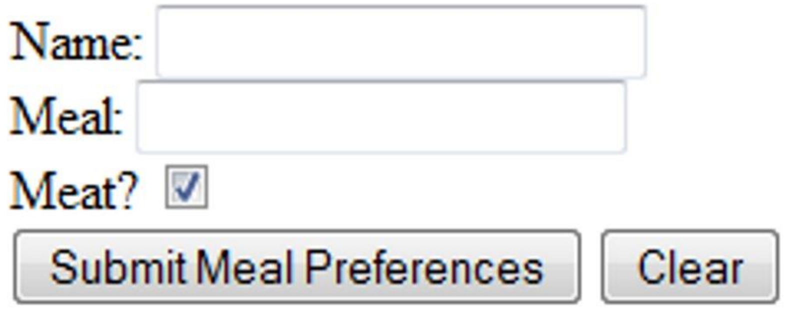
```
<select name="favoritecharacter">
  <optgroup label="Major Characters">
    <option>Jerry</option>
    <option>George</option>
    <option>Kramer</option>
    <option>Elaine</option>
  </optgroup>
  <optgroup label="Minor Characters">
    <option>Newman</option>
    <option>Susan</option>
  </optgroup>
</select>
```

Reset buttons

when clicked, returns all form controls to their initial values

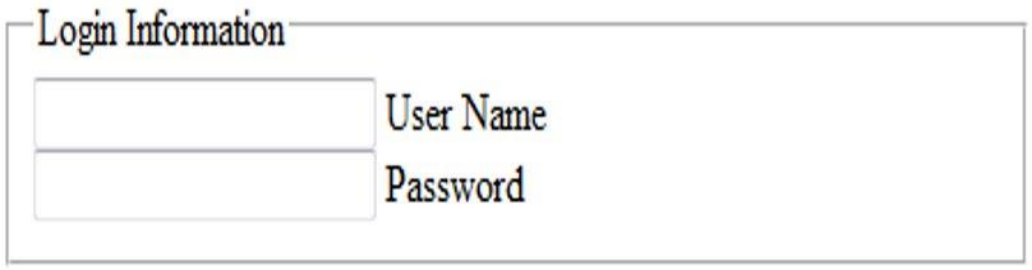
```
<form>
  <div>
    <label>Name: <input type="text" name="name" /></label> <br />
    <label>Meal: <input type="text" name="meal" /></label> <br />
    <label>Meat?
      <input type="checkbox" name="meat" checked="checked" />
    </label> <br />

    <input type="submit" value="Submit Meal Preferences" />
    <input type="reset" value="Clear" />
  </div>
</form>
```



Grouping input: <fieldset>, <legend>

- groups of input fields with optional caption (block)
- `fieldset` groups related input fields
- `legend` supplies an optional caption


Login Information

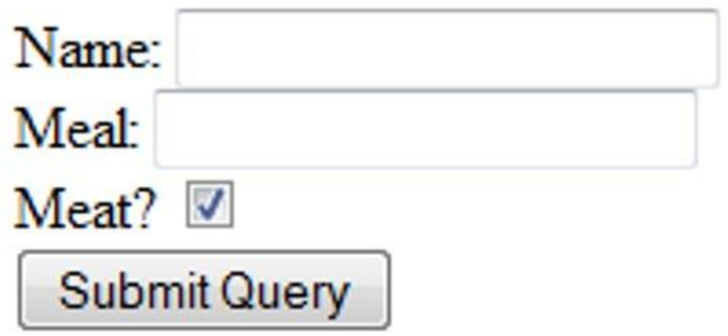
User Name

Password

```
<form action= "" >
  <fieldset>
    <legend>Login Information</legend>
    <input type="text" name="username" /> User Name <br />
    <input type="text" name="password" /> Password
  </fieldset>
</form>
```

Hidden input parameters

- an invisible parameter that is still passed to the server when form is submitted
- useful for passing on additional state that isn't modified by the user



```
<form action="">
  <fieldset>
    <label>Name: <input type="text" name="name"
/></label> <br />
    <label>Meal: <input type="text" name="meal"
/></label> <br />
    <label>Meat?
      <input type="checkbox" name="meat"
checked="checked" />
    </label> <br />

    <!-- two hidden input parameters -->
    <input type="hidden" name="organization"
value="CocaCola" />
    <input type="hidden" name="year" value="2019"
/>

    <input type="submit" />
  </fieldset>
</form>
```

HTML Images

The <img> tag is empty, it contains attributes only, and does not have a closing tag.

The src attribute specifies the URL (web address) of the image

```
<img src="url">
```

Images Paths

- related

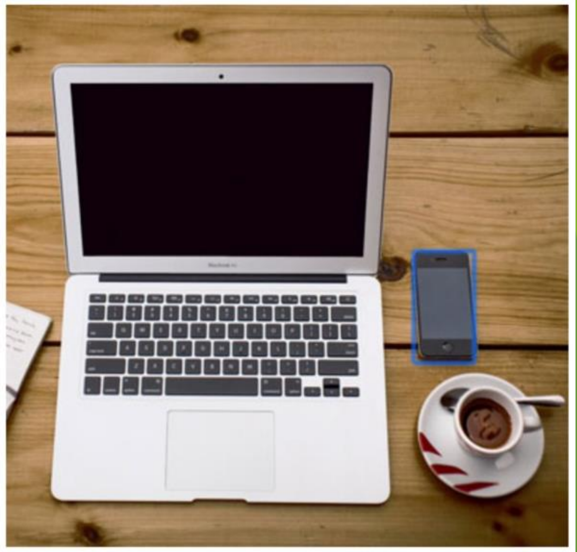
<img src="/images/html5.gif" alt="HTML5 Icon"
  style="width:128px;height:128px;"\>

- absolute

<img src="https://www.w3schools.com/images/w3schools_
  green.jpg" alt="W3Schools.com"\>

Image Maps

- The <map> tag defines an image-map. An image-map is an image with clickable areas.
- In the image near, click on the computer, the phone, or the cup of coffee
- The name attribute of the <map> tag is associated with the <img>'s usemap attribute and creates a relationship between the image and the map
- The <map> element contains a number of <area> tags, that define the clickable areas in the image-map



```
<img src="workplace.jpg" alt="Workplace" usemap="#workmap">
<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer”
       href="computer.htm">
  <area shape="rect" coords="290,172,333,250" alt="Phone"
       href="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Coffee"
       href="coffee.htm">
</map>
```

The <picture> Element

- HTML5 introduced the <picture> element to add more flexibility when specifying image resources.
- The <picture> element contains a number of <source> elements, each referring to different image sources. This way the browser can choose the image that best fits the current view and/or device.
- Each <source> element have attributes describing when their image is the most suitable.
- The browser will use the first <source> element with matching attribute values, and ignore any following <source> elements.

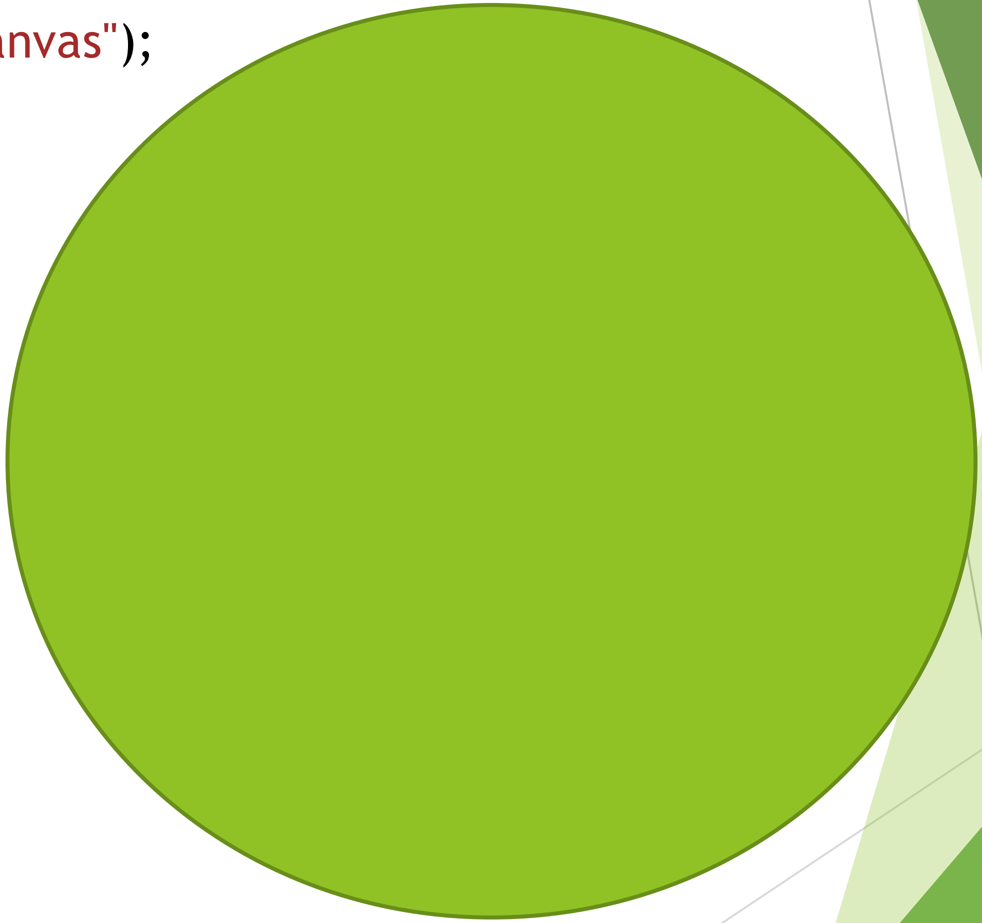
```
<picture>
  <source media="(min-width: 650px)" srcset="img_pink_flowers.jpg">
  <source media="(min-width: 465px)" srcset="img_white_flower.jpg">
  <img src="img_orange_flowers.jpg" alt="Flowers" style="width:auto">
</picture>
```

Canvas

- The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript.
- The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics.
- Canvas has several methods for drawing paths, boxes, circles, text, and adding images.
- A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

```
const c = document.getElementById("myCanvas");
const ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95, 50, 40, 0, 2 * Math.PI);
ctx.stroke();
```



SVG

- SVG stands for Scalable Vector Graphics
- SVG is used to define graphics for the Web
- SVG is a W3C recommendation

-
  ```
  <svg width="100" height="100">
    <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="lightgreen" />
  </svg>
  ```

Differences Between SVG and Canvas

SVG is a language for describing 2D graphics in XML.

Canvas draws 2D graphics, on the fly (with a JavaScript).

SVG is XML based, which means that every element is available within the SVG DOM. You can attach JavaScript event handlers for an element.

In SVG, each drawn shape is remembered as an object. If attributes of an SVG object are changed, the browser can automatically re-render the shape.

Canvas is rendered pixel by pixel. In canvas, once the graphic is drawn, it is forgotten by the browser. If its position should be changed, the entire scene needs to be redrawn, including any objects that might have been covered by the graphic.

HTML Audio

- The controls attribute adds audio controls, like play, pause, and volume.
- The <source> element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.
- The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
Your browser does not support the
audio element.
</audio>
```

Videos in HTML

- Before HTML5, a video could only be played in a browser with a plug-in (like flash).
- The HTML5 <video> element specifies a standard way to embed a video in a web page.

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

How it Works

The controls attribute adds video controls, like play, pause, and volume.

It is a good idea to always include width and height attributes.

If height and width are not set, the page might flicker while the video loads.

The <source> element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.

The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

HTML <meta> tags

Define the character set used:	• <meta charset="UTF-8">
Define a description of your web page:	• <meta name="description" content="Free Web tutorials">
Define keywords for search engines:	• <meta name="keywords" content="HTML, CSS, XML, JavaScript">
Define the author of a page:	• <meta name="author" content="John Doe">
Refresh document every 30 seconds:	• <meta http-equiv="refresh" content="30">
Setting The Viewport	• <meta name="viewport" content="width=device-width, initial-scale=1.0">

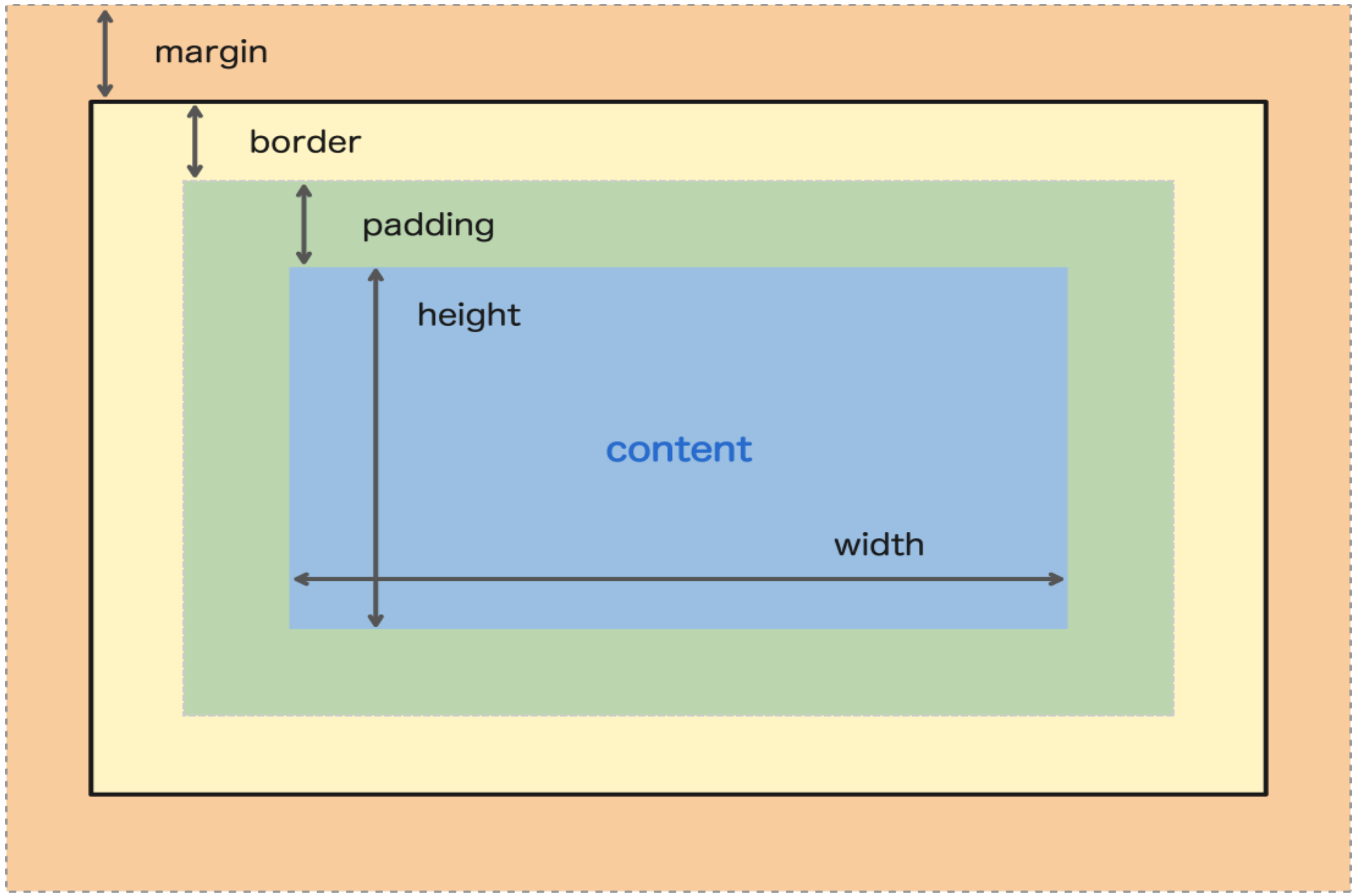
Best Practices

- Always keep your code tidy, clean, and well-formed
- Use Lower Case File Names
- Use Correct Document Type
- Use Lower Case Element Names
- Close All HTML Elements
- Close Empty HTML Elements
- Use Lower Case Attribute Names
- Quote Attribute Values
- Image Attributes – always use alt attribute
- Don't' Use Spaces Between Equal Signs
- Avoid Long Code Lines (no longer than 80 - 120 characters)
- Do Not Add Blank Lines Without a Reason
- Don't Omit the <html>, <head> and the <body> Tag
- Specify the page language in <html> tag
- <title> tag is required
- Use <meta> tags
- Setting The Viewport
- Put HTML Comments

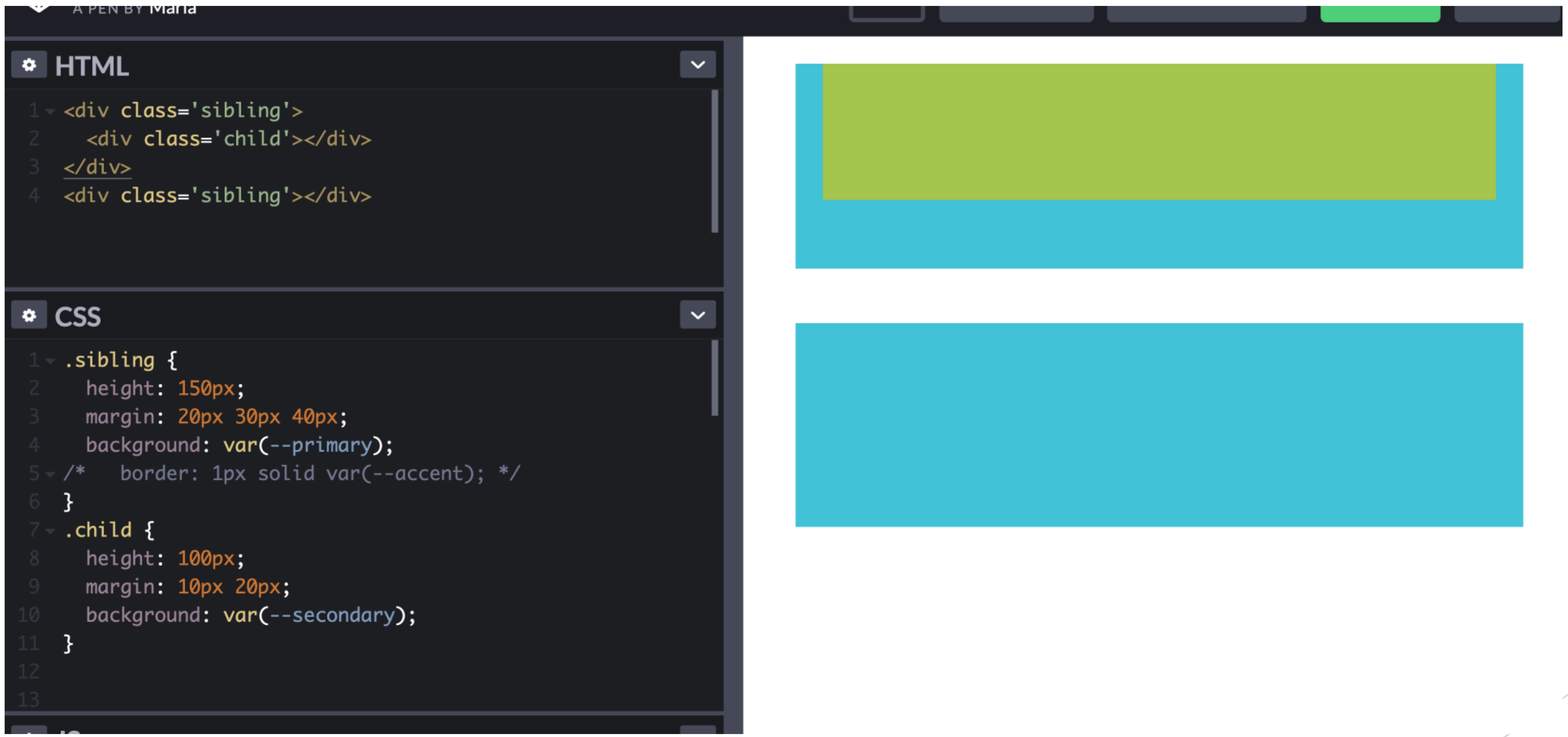
CSS LAYOUT

General page blocks organization which defines structure of a page and positioning of its content.

Box Model



Margin Collapse


A PEN BY Maria

HTML

```html
<div class='sibling'>
  <div class='child'></div>
</div>
<div class='sibling'></div>
```

CSS

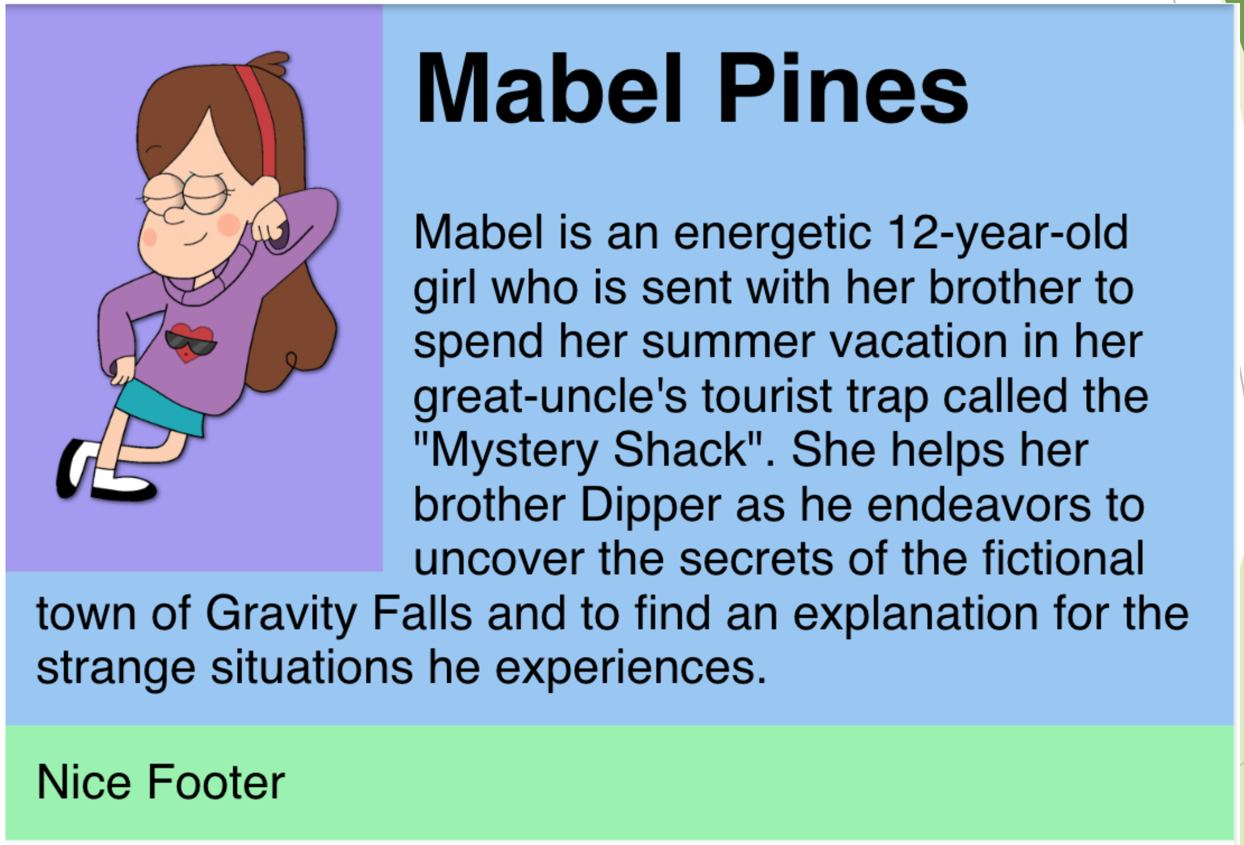
```css
.sibling {
  height: 150px;
  margin: 20px 30px 40px;
  background: var(--primary);
/*   border: 1px solid var(--accent); */
}
.child {
  height: 100px;
  margin: 10px 20px;
  background: var(--secondary);
}
```

Normal flow

Floats & clear-fix

- For old browsers
- To set the flow around the right or left
- Clear-fix

```
.clearfix::after {
    content: "";
    display: table;
    clear: both;
}
```



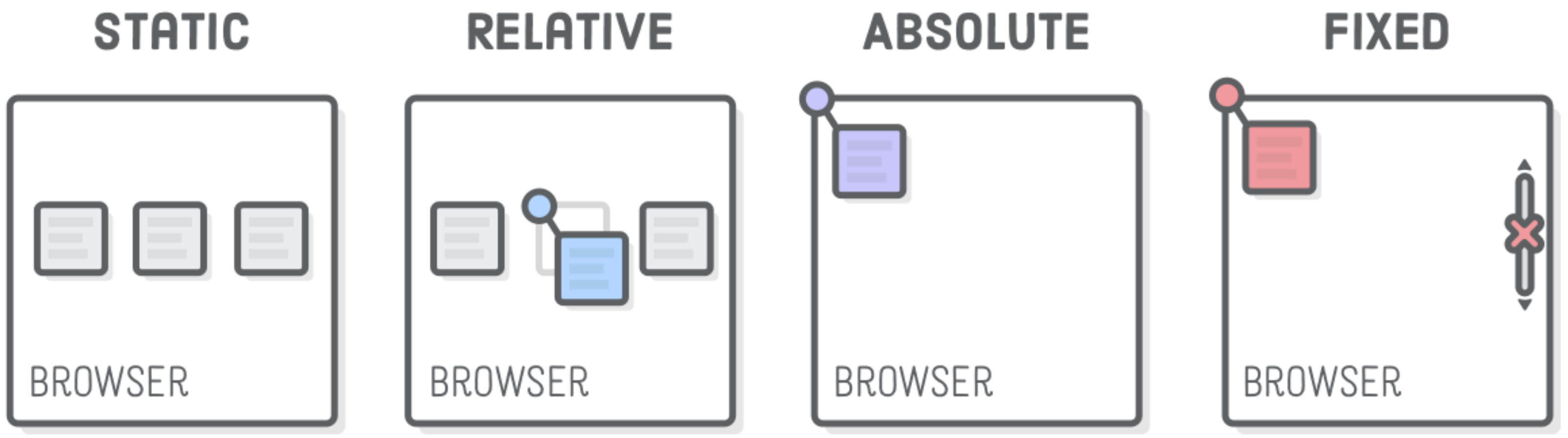
Mabel Pines

Mabel is an energetic 12-year-old girl who is sent with her brother to spend her summer vacation in her great-uncle's tourist trap called the "Mystery Shack". She helps her brother Dipper as he endeavors to uncover the secrets of the fictional town of Gravity Falls and to find an explanation for the strange situations he experiences.

Nice Footer

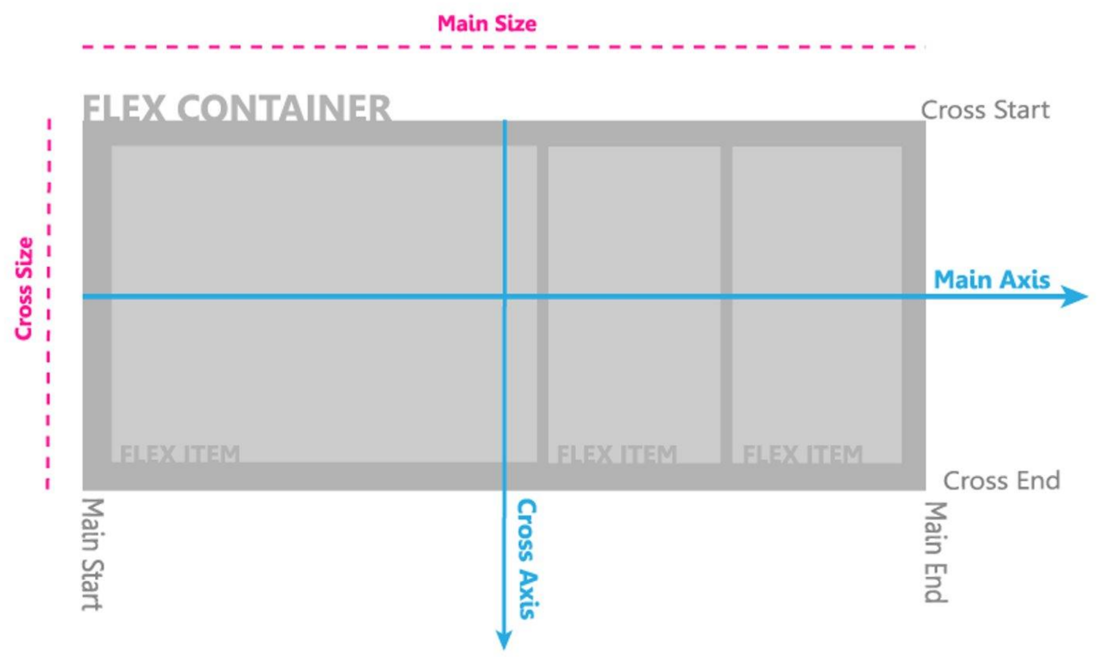
Position

- static | relative | absolute | sticky | fixed
- The 'top', 'right', 'bottom', and 'left' properties do not apply for position: static

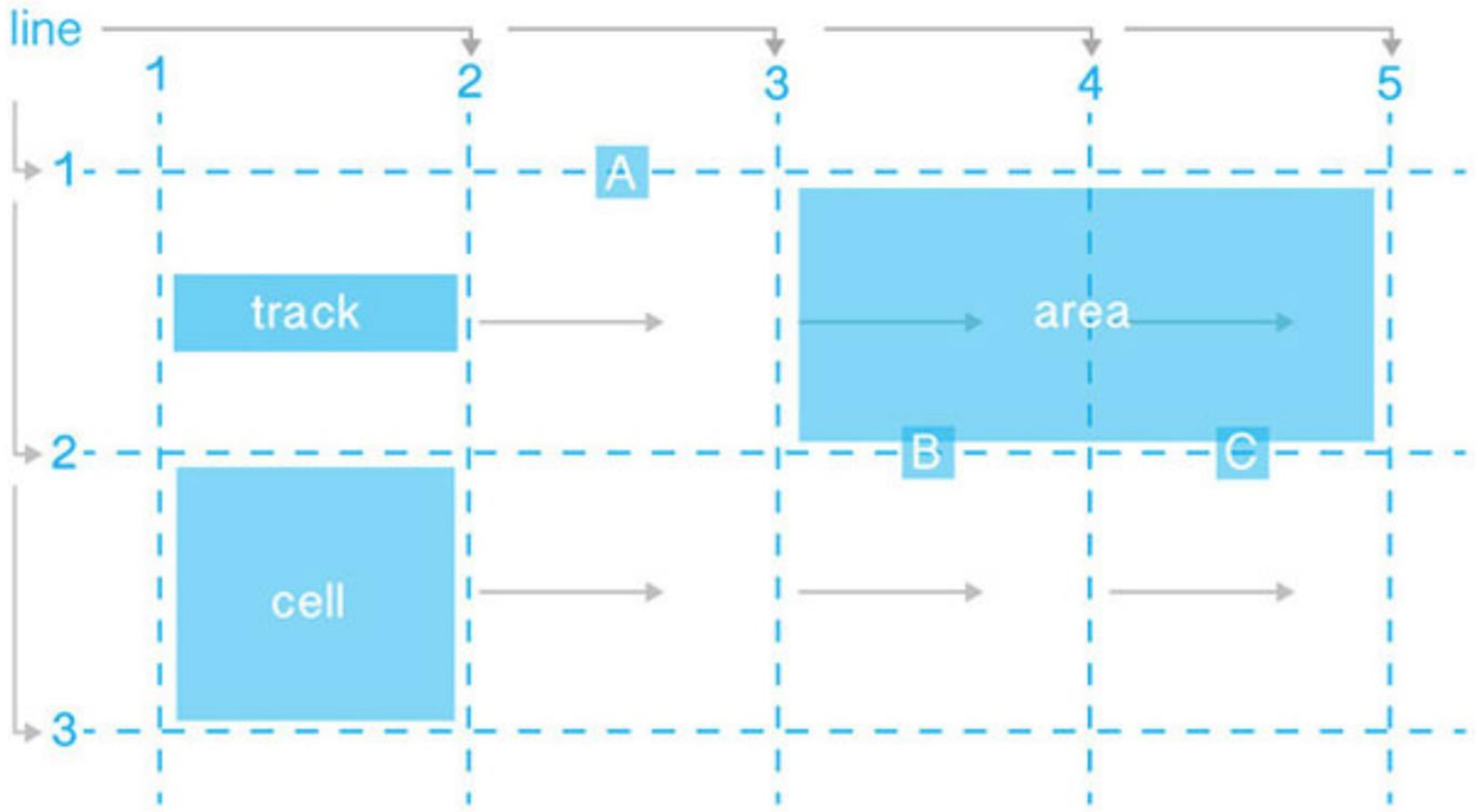


Flexbox

- **Main axis** - main axis of a flex container is the primary axis along which flex items are laid out
- **Cross axis** - axis perpendicular to the main axis



Grid

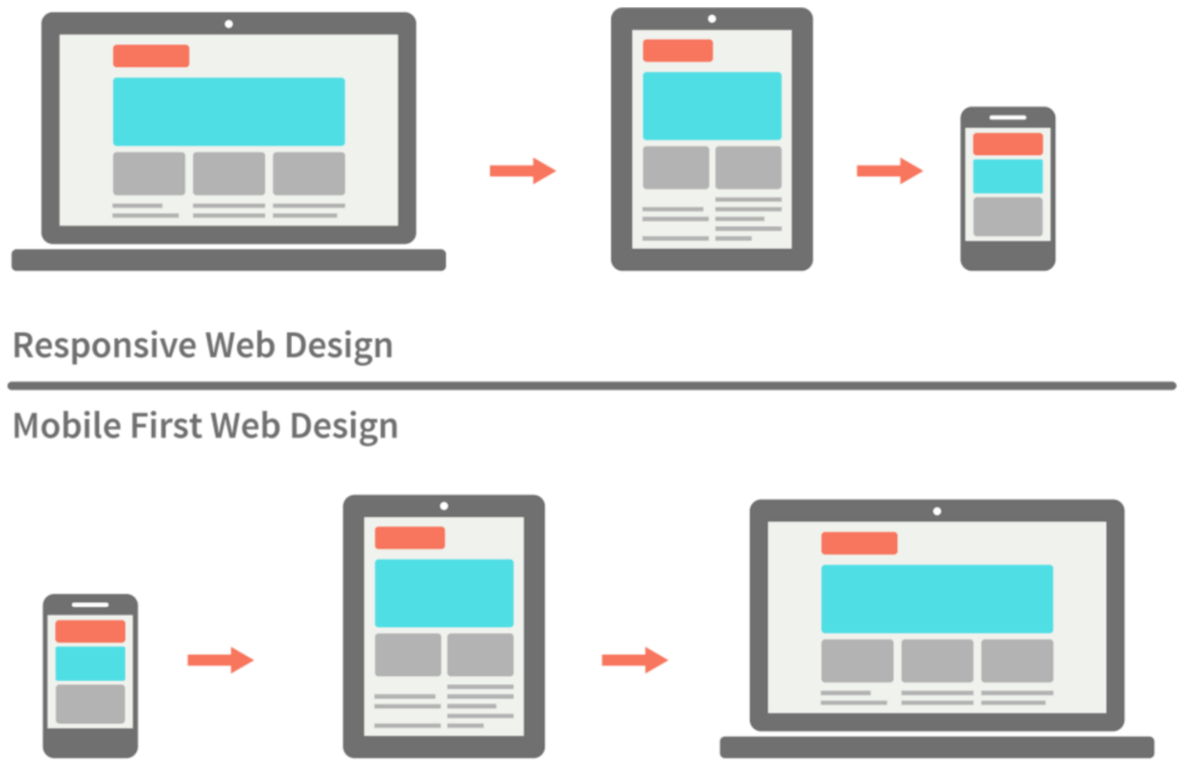


DEFINITIONS

- **Grid container** - element on which display: grid is applied
- **Grid item** - children (i.e. direct descendants) of the grid container
- **Grid line** - dividing lines that make up the structure of the grid
- **Grid track** - space between two adjacent grid lines
- **Grid cell** - space between two adjacent row and two adjacent column grid lines
- **Grid area** - total space surrounded by four grid lines
- **Fraction (fr)** - fraction of the free space of the grid container; calculated after any non-flexible items

MAIN STEP TO RESPONSIVE

- Fluid grids
- Flexible images
- Media queries



Fluid grids

01

Define block sizes in %, vw, vh

02

Convert font sizes from px to rem (em)

03

Use grid and flexbox

Flexible images

- max-width: 100%
- HTML5 <picture> tag and srcset
- Vector graphic

MEDIA QUERY

target CSS rules based on screen size, device-orientation or display-density etc.

```
@media [feature] ([condition]) { CSS rule }
```

```
@media all and (min-width: 768px) { … }
```

```
@media (orientation: portrait) { … }
```

```
@media only screen and (min-resolution: 192dpi) { … }
```

```
@media screen and (device-width: 800px) { … }
```

```
@media (min-device-pixel-ratio: 2) { … }
```

- Thank You!