

UI Programming

Let's enter into the Web world

UI Design Principles

Make User Interface consistent

Allow users to navigate easily

Provide informative feedback

Design dialog to yield closure

Prevent error as much as possible

Allow users to reverse their actions easily

Minimize memory load

Clarity

Familiarity

Role of UI Developer

A UI developer's role is to translate creative software design concepts and ideas into reality using frontend technologies.

UI Developer Skills

In depth knowledge of user interface development tech stack and frameworks

- HTML
- CSS
- JavaScript
- Angular
- ReactJS
- Typescript
- AJAX etc

A level of design and business understanding so as to not compromise on business goals and design intents

UX Designer vs UI Developer

Create user stories, persona and storyboards

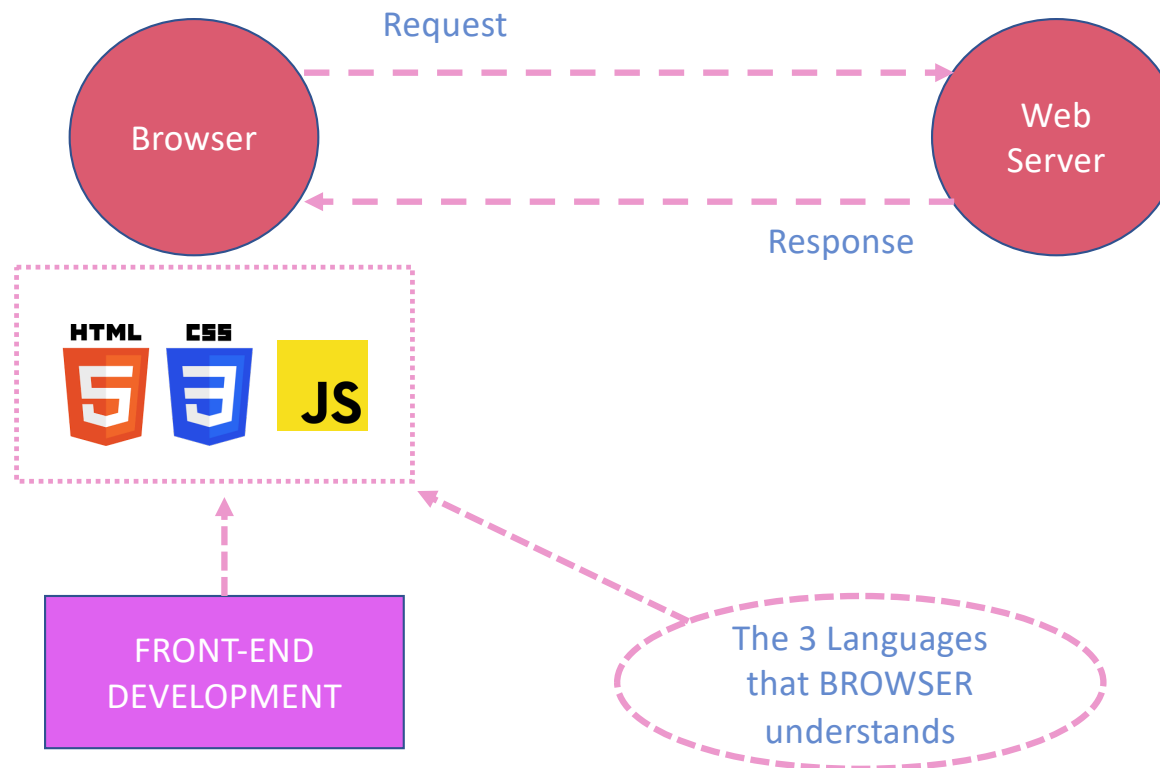
Determine information architecture and create sitemaps

Create prototypes and wireframes

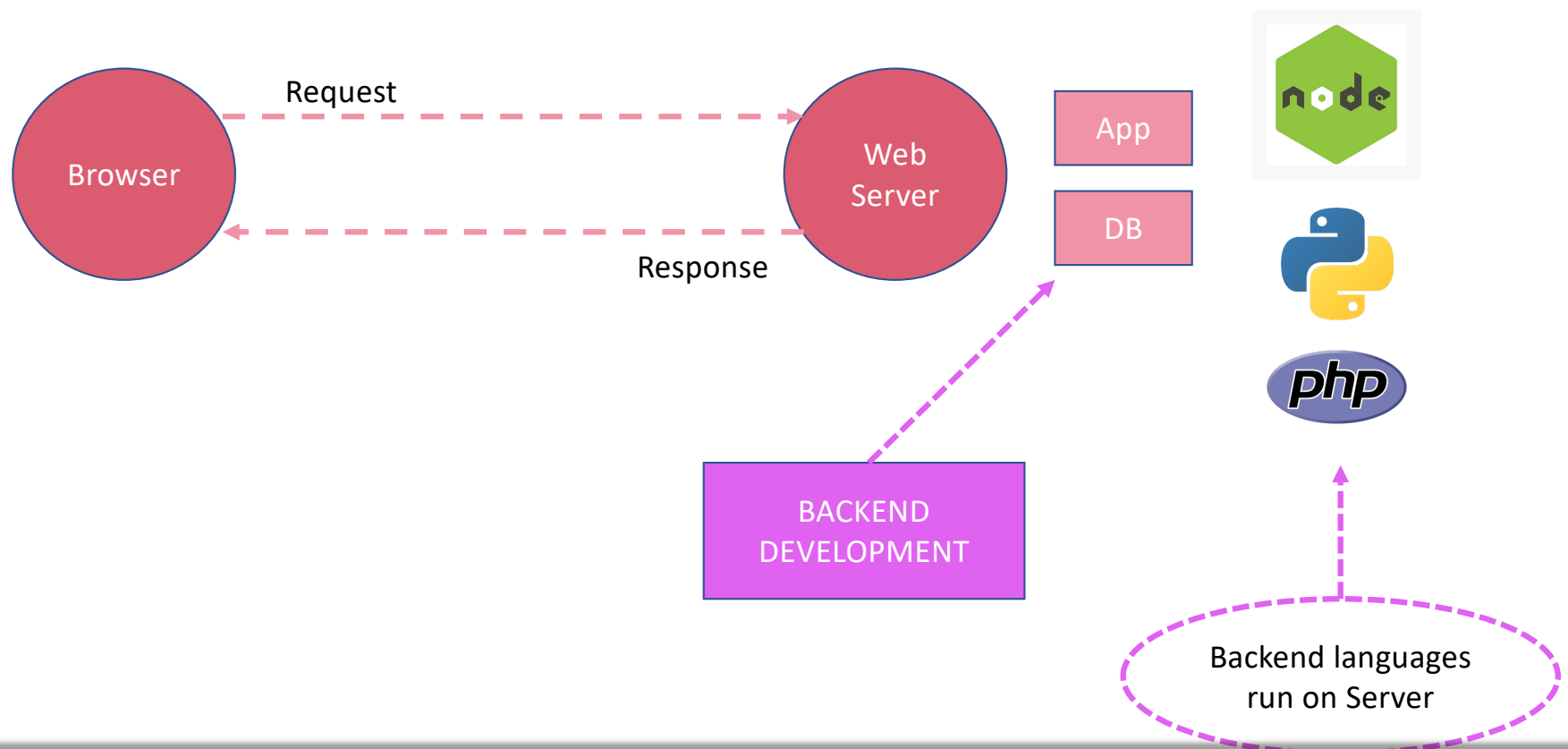
Conduct usability testing

Frontend Technologies

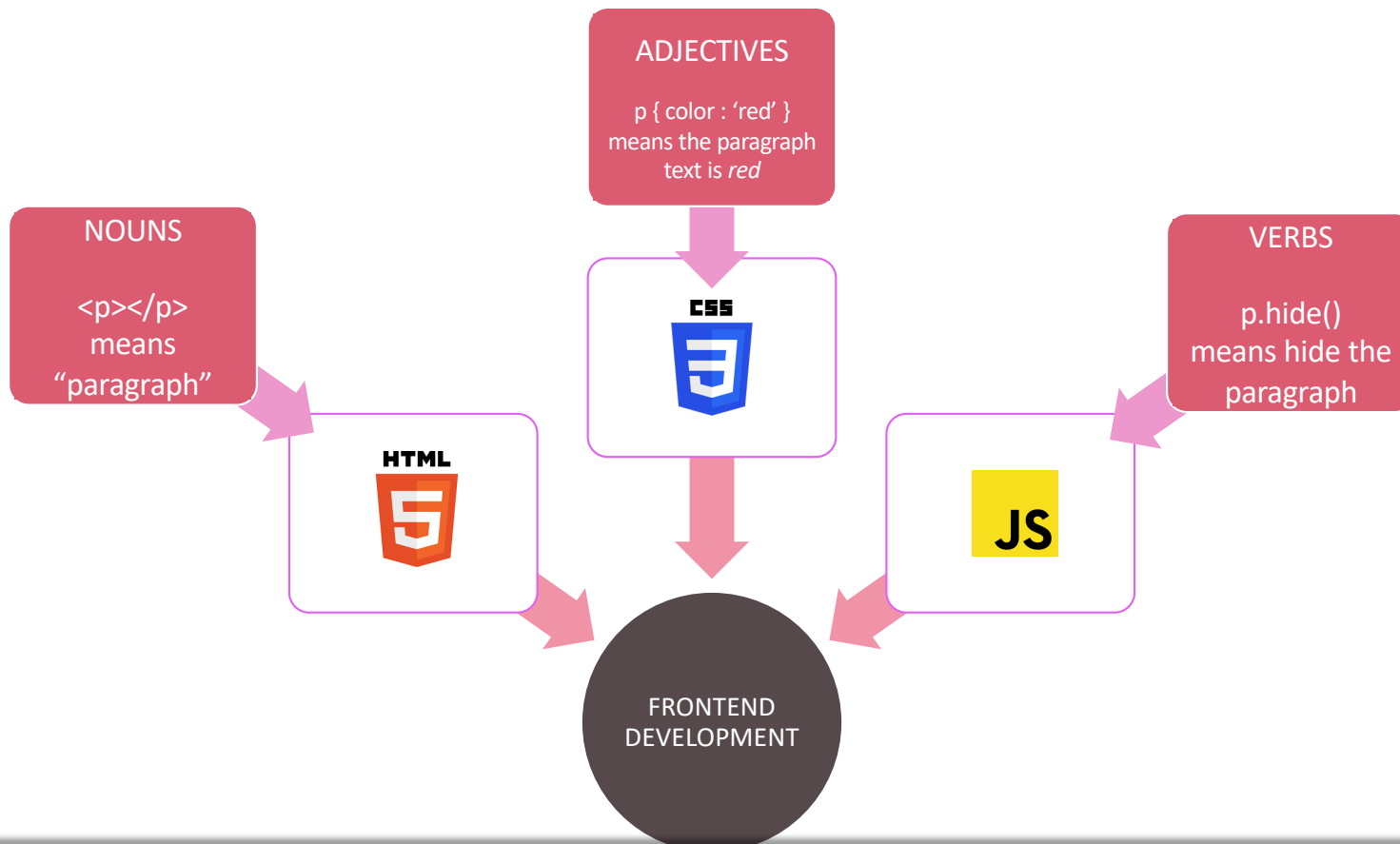
FRONTEND DEVELOPMENT



BACKEND DEVELOPMENT



THE 3 LANGUAGES OF FRONTEND



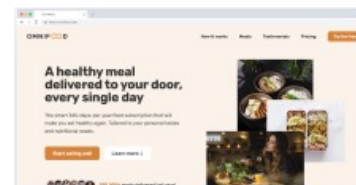
INTRODUCTION TO HTML

HyperText Markup Language

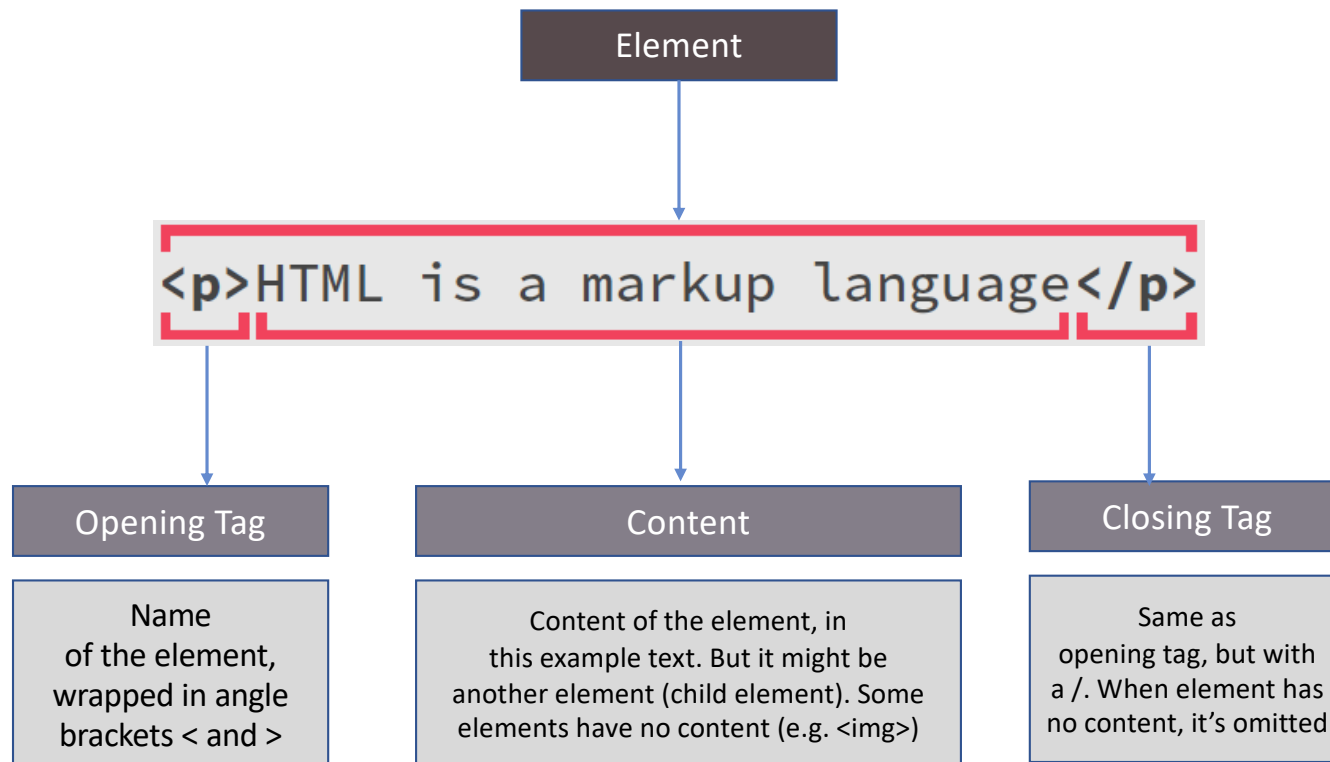
HTML is a markup language that web developers use to structure and describe the content of a webpage (not a programming language)

HTML consists of elements that describe different types of content: paragraphs, links, headings, images, video, etc.

Web browsers understand HTML and render HTML code as websites



ANATOMY OF AN HTML ELEMENT



HTML Elements

| | |
|--------|---|
| title | specifies the label that appears in the browser window's <i>title bar</i> . |
| meta | Provide information about the web page. |
| style | Writing internal stylesheet for your page. |
| script | Adding actions to your app by including JavaScript. |
| body | Includes various other elements to display data on webpage |

CSS

Let's make it better looking

INTRODUCTION TO CSS

Cascading Style Sheets

CSS describes the visual style and presentation of the content written in HTML

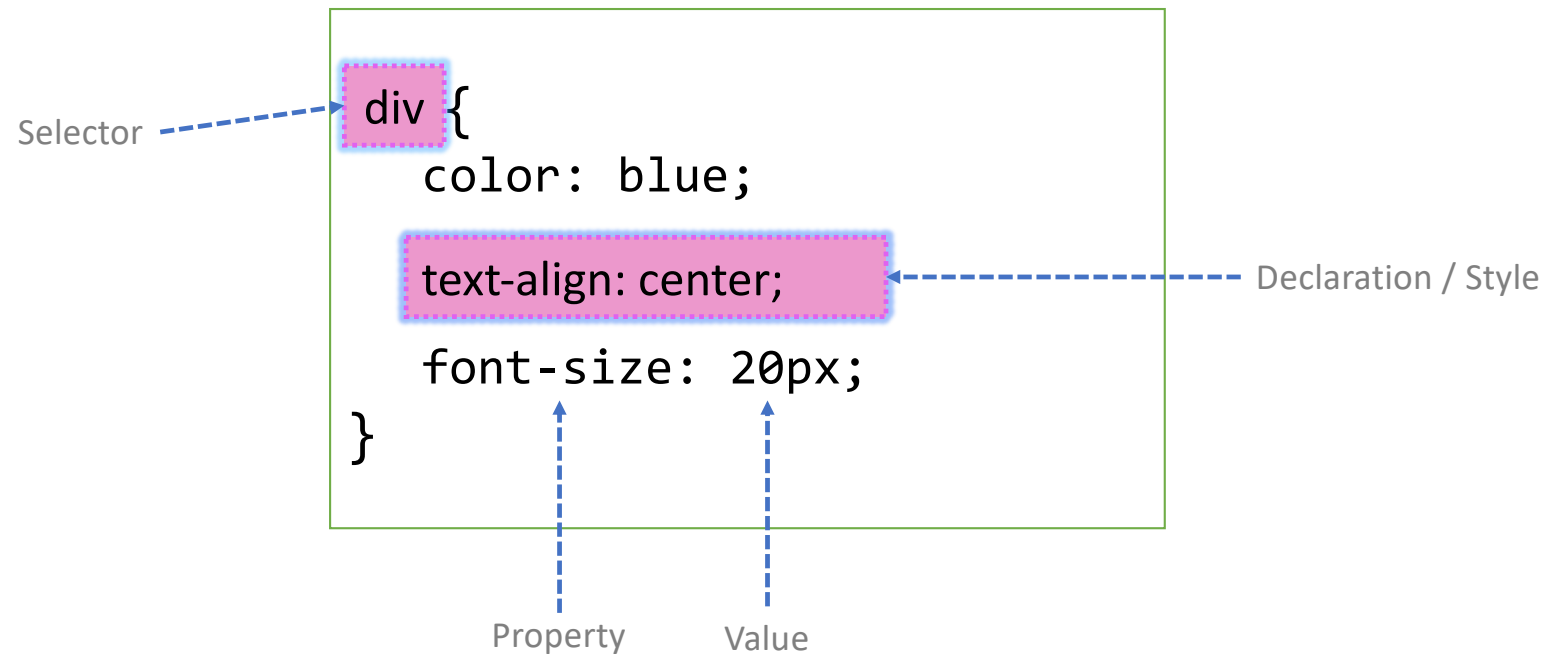
CSS consists of countless properties that developers use to format the content: properties about font, text, spacing, layout, etc.

Web browsers understand HTML and render HTML code as websites



HOW WE SELECT AND STYLE ELEMENTS

A CSS RULE



CSS SELECTORS

| Selector Type | Syntax | Description | Example |
|---------------|-------------|---|---|
| Universal | * | Selects all elements | * will match all the elements |
| Type | elementName | Selects all elements that have the given node name | input will match any <input> element. |
| Class | .classname | Selects all elements that have the given class attribute | .index will match any element that has a class of "index" |
| ID | #id | Selects an element based on the value of its id attribute | #toc will match the element that has the ID "toc" |
| Attribute | [attr] | Selects all elements that have the given attribute | [autoplay] will match all elements that have the autoplay attribute |

MORE CSS SELECTORS

| Selector Type | Syntax | Description | Example |
|-----------------------------|--------|--|--|
| Grouping Selector List | , | The , selector is a grouping method that selects all the matching nodes | div, span will match both and <div> elements |
| Descendant combinator | " " | The " " (space) combinator selects nodes that are descendants of the first element | div span will match all elements that are inside a <div> element |
| Child combinator | > | The > combinator selects nodes that are direct children of the first element | ul > li will match all elements that are nested directly inside a element. |
| General sibling combinator | ~ | The ~ combinator selects siblings. This means that the second element follows the first (though not necessarily immediately), and both share the same parent | p ~ span will match all elements that follow a <p>, immediately or not. |
| Adjacent sibling combinator | + | The + combinator matches the second element only if it immediately follows the first element | h2 + p will match all <p> elements that immediately follow an <h2> element |

MORE CSS SELECTORS

| Selector Type | Syntax | Description | Example |
|-----------------|--------|--|--|
| Pseudo classes | : | The : pseudo allow the selection of elements based on state information that is not contained in the document tree | a:visited will match all <a> elements that have been visited by the user |
| Pseudo elements | :: | The :: pseudo represent entities that are not included in HTML | p::first-line will match the first line of all <p> elements |

WHAT IS RESPONSIVE DESIGN?

Design technique to make a webpage adjust its layout and visual style to **any possible screen size** (window or viewport size)

In practice, this means that responsive design makes websites usable on all devices, such as **desktop computers, tablets, and mobile phones**.

It's a set of practices, **not a separate technology**. It's all just CSS!

RESPONSIVE DESIGN INGREDIENTS

FLUID LAYOUTS

To allow webpage to adapt to the current viewport width (or even height)

Use % (or vh / vw) unit instead of px for elements that should adapt to viewport (usually layout)

Use **max-width** instead of width

RESPONSIVE UNITS

Use **rem** unit instead of px for most lengths to make it easy to scale the entire layout down (or up) automatically

Helpful trick: setting 1rem to 10px for easy calculations

FLEXIBLE IMAGES

By default, images don't scale automatically as we change the viewport, so we need to fix that

Always use % for image dimensions, together with the **max-width** property

Use max-width instead of width

MEDIA QUERIES

Bring responsive sites to life!

To change CSS styles on certain viewport widths (called breakpoints)

Use media queries and select breakpoints

DESKTOP-FIRST VS. MOBILE-FIRST DEVELOPMENT

DESKTOP-FIRST

Start writing CSS for the desktop: large screen

Then, media queries shrink design to smaller screens.

MOBILE-FIRST

Start writing CSS for mobile devices: small screen

Then, media queries expand design to a large screen

Forces us to reduce websites and apps to the absolute essentials.

WHAT IS BOOTSTRAP?

Bootstrap is an open source product from Mark Otto and Jacob Thornton who were both employees at Twitter.

It is sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development.

WHY BOOTSTRAP ?

Mobile first
approach

Browser
Support

Easy to get
started

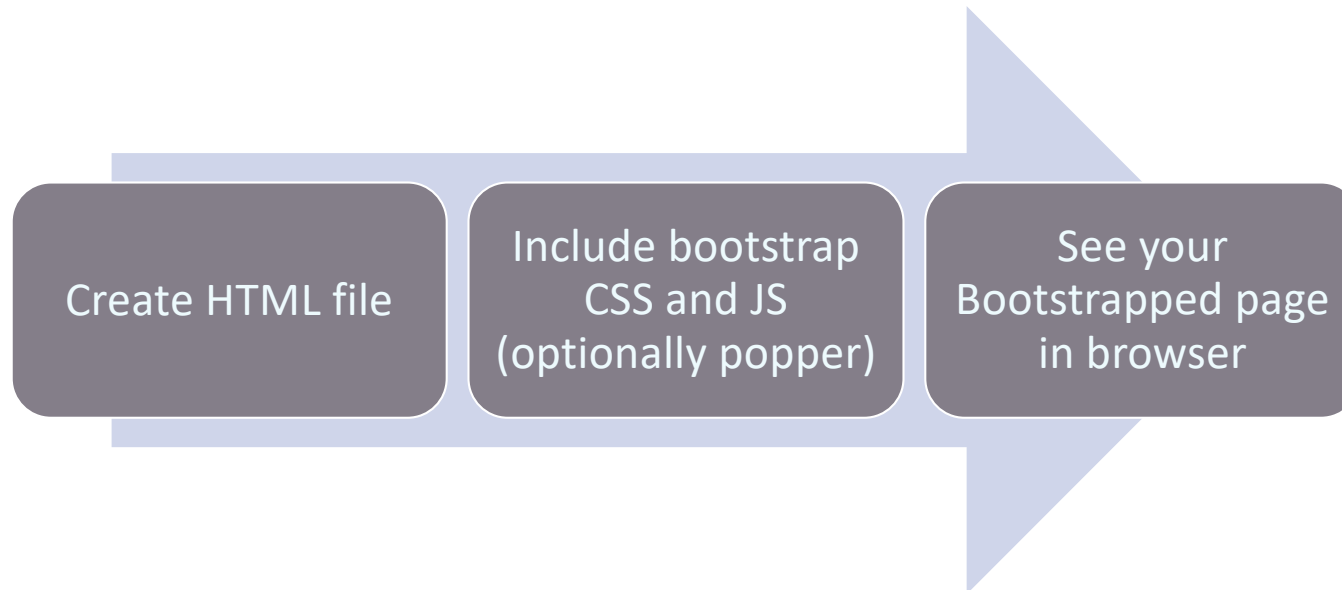
Responsive
design

Uniform
solution for
building UI

Beautiful and
functional built-
in components

LET'S GET OUR HANDS DIRTY

You can include Bootstrap's production-ready CSS and JavaScript via CDN without the need for any build steps.



BOOTSTRAP FUNCTIONALITIES

Layout

- Breakpoints
- Container
- Grids
- Columns / Gutter

Contents

- Colors
- Typography
- Images
- Tables

Forms

- Form Controls
- Checks & Radios
- Input Groups & Floating Labels
- Form Validations

Components

- Buttons
- Alerts
- Progress Bar
- Pagination
- Navbar
- Modal

Utilities

- Vector Icons
-

LAYOUT : BREAKPOINTS

Breakpoints are customizable widths that determine how your responsive layout behaves across device or viewport sizes in Bootstrap

| Breakpoint | Class infix | Dimensions |
|-------------------|-------------|------------|
| Extra Small | None | <576px |
| Small | sm | ≥576px |
| Medium | md | ≥768px |
| Large | lg | ≥992px |
| Extra Large | xl | ≥1200px |
| Extra Extra Large | xxl | ≥1400px |

LAYOUT : GRID SYSTEM

Powerful mobile-first flexbox grid to build layouts of all shapes and sizes with twelve column system

Bootstrap's grid system can adapt across all six default breakpoints, and any breakpoints you customize.

How the
grid
system
comes
together -

Grid supports six responsive breakpoints

Containers center and horizontally pad your content

Rows are wrappers for columns

Columns are incredibly flexible

Gutters are also responsive and customizable

Sass variables, maps, and mixins power the grid

LAYOUT : COLUMNS

You can modify columns with a handful of options for alignment, ordering, and offsetting thanks to flexbox grid system.

You can also use column classes to manage widths of non-grid elements

How the Columns works -

Columns build on the grid's flexbox architecture

When building grid layouts, all content goes in columns

Bootstrap includes predefined classes for creating fast, responsive layouts

JAVASCRIPT

THE SCRIPTING LANGUAGE FOR WEB PROGRAMMING

WHAT IS JAVASCRIPT?

JAVASCRIPT IS A HIGH-LEVEL,
OBJECT-ORIENTED, MULTI-PARADIGM
PROGRAMMING LANGUAGE.

We don't have to worry about complex stuff
like memory management

We can use different styles
of programming

Based on objects,
for storing most kinds of data

Instruct computer to do things

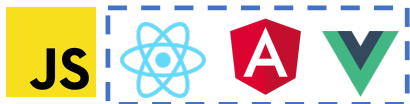
JAVASCRIPT FEATURES

| | | |
|--|------------------------------------|-----------------------|
| HIGH-LEVEL | PROTOTYPE-BASED OBJECT-ORIENTED | MULTI-PARADIGM |
| INTERPRETED OR JUST-IN-TIME COMPILED | DYNAMIC | SINGLE-THREADED |
| NON-BLOCKING EVENT LOOP | FIRST-CLASS FUNCTIONS | GARBAGE- COLLECTED |

THERE IS NOTHING YOU CAN'T DO WITH JAVASCRIPT

FRONT-END APPS

Dynamic effects and
web applications in the
browser



100% based on JavaScript.
They might go away,
but JavaScript won't!

Native mobile
applications



BACK-END APPS

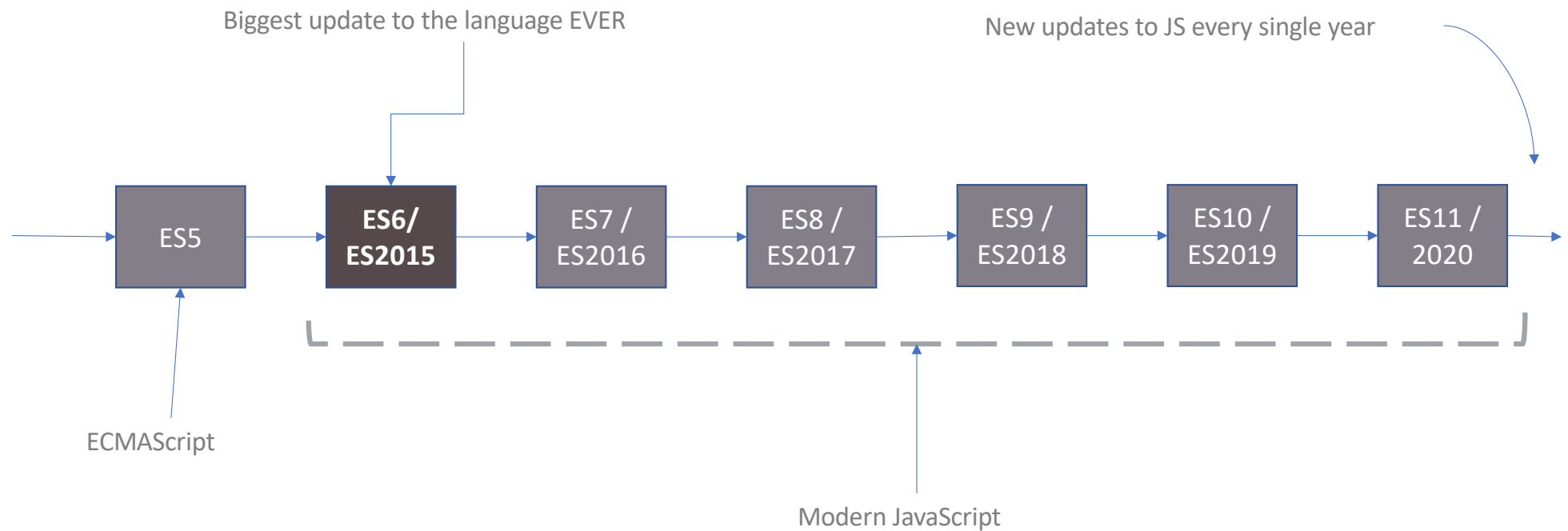
Web applications on
web servers



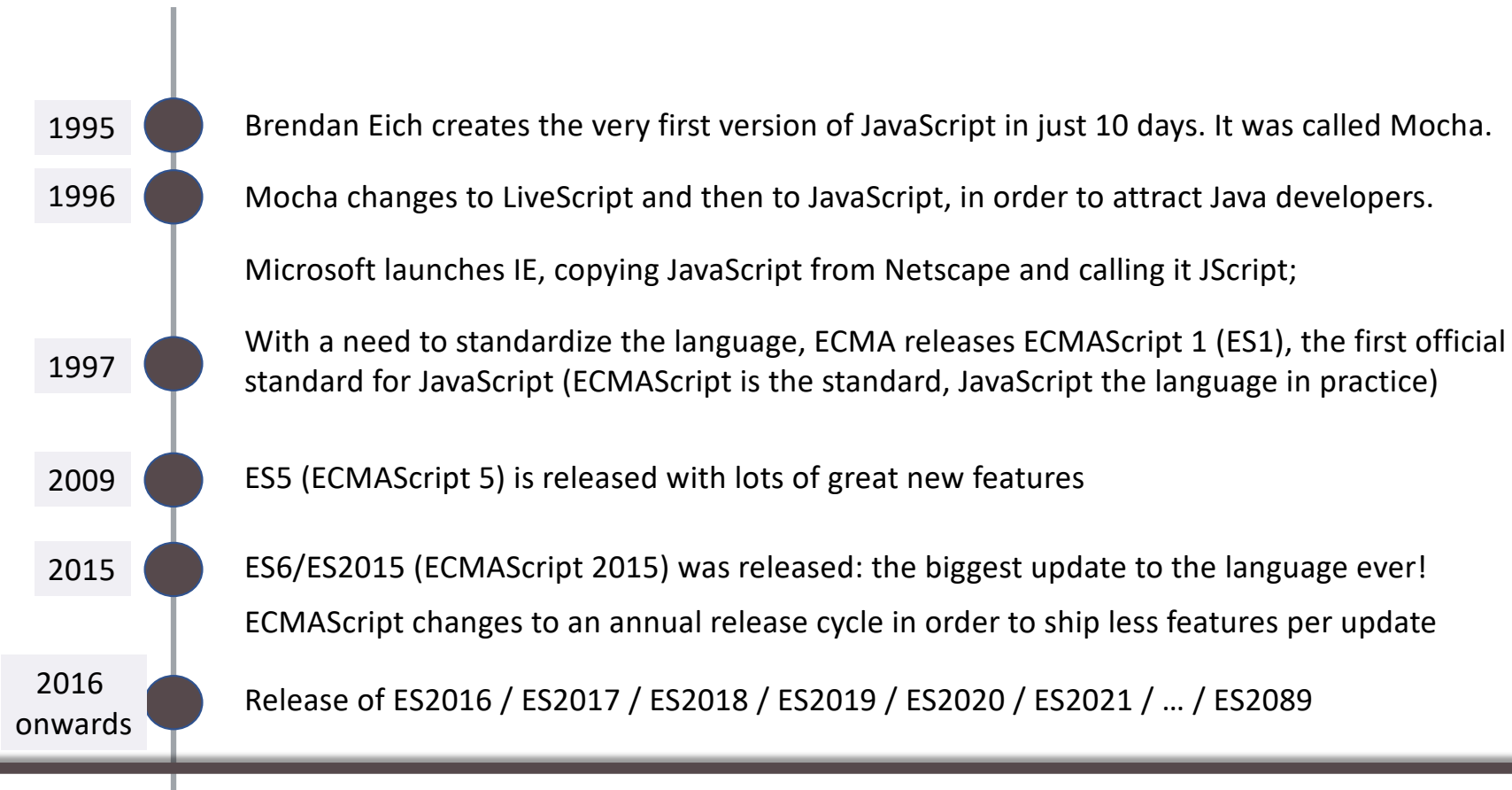
Native desktop
applications



JAVASCRIPT RELEASES...



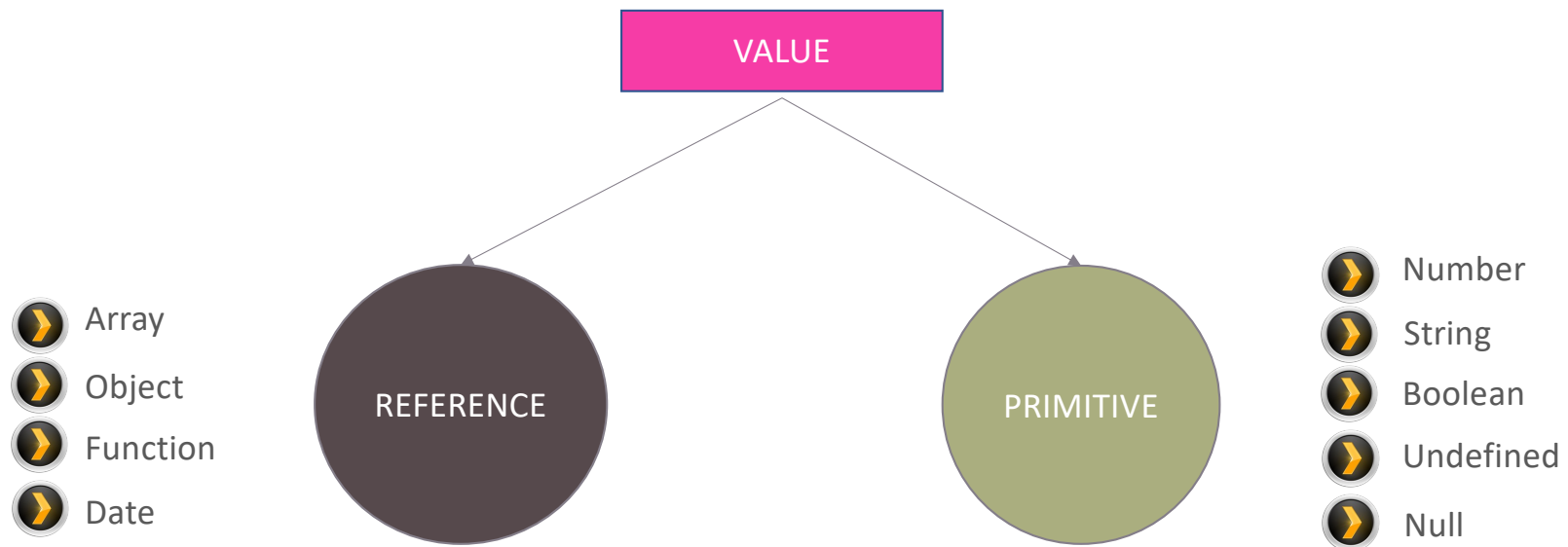
A BRIEF HISTORY OF JAVASCRIPT



DATA TYPES

LET'S GIVE IT A TYPE

OBJECTS AND PRIMITIVES



DATA STRUCTURE

STORING DATA IN OBJECT AND ARRAY

OBJECTS & ARRAYS

Objects represented by Flower Brackets – { }

- The Object type represents one of JavaScript's data types
- It is used to store various keyed collections and more complex entities
- Objects can be created using the Object() constructor or the object initializer / literal syntax

Arrays represented by Square Bracket - []

- The Array object, as with arrays in other programming languages, enables storing a collection of multiple items under a single variable name, and has members for performing common array operations

WORKING WITH ARRAY

A COLLECTION OF MULTIPLE ITEMS UNDER A SINGLE VARIABLE NAME

ARRAY IN JAVASCRIPT

The Array object, as with arrays in other programming languages, enables storing a collection of multiple items under a single variable name, and has members for performing common array operations.

JavaScript arrays are resizable and can contain a mix of different data types

JavaScript arrays are not associative arrays and so, array elements cannot be accessed using strings as indexes

JavaScript arrays are zero-indexed

JavaScript array-copy operations create shallow copies

WHICH ARRAY METHOD TO USE?

I WANT ...

TO MUTATE ORIGINAL ARRAY

- Add methods -
 - push()
 - unshift()
- Remove methods -
 - Pop()
 - shift()
 - Splice()
- Others -
 - Reverse()
 - Sort()
 - Fill()

A NEW ARRAY

- Computed from original-
 - Map()
- Filtered using condition-
 - Filter()
- Portion of original -
 - Slice()
- Adding original to other-
 - Concat()

AN ARRAY INDEX

- Based on value-
 - indexOf()
- Based on test condition
 - findIndex()
- An array element -
 - Find()

WHICH ARRAY METHOD TO USE?

I WANT ...

KNOW IF ARRAY INCLUDES

- Based on value -
 - Includes()
- Based on test condition -
 - some()
 - every()
- Based on separator string-
 - Join()

TO TRANSFORM TO VALUE

- Based on accumulator
 - reduce()
- Based on callback-
 - forEach()

FUNCTIONS

FIRST CLASS CITIZENS IN JAVASCRIPT

FUNCTIONS

Function declaration

Function that can be used before it's declared

```
function calcAge(birthYear) {  
  return 2037 - birthYear;  
}
```

Function expression

Essentially a function value stored in a variable

```
const calcAge = function (birthYear) {  
  return 2037 - birthYear;  
};
```

Arrow function

Great for a quick one-line functions. (more later...)

```
const calcAge = birthYear => 2037 - birthYear;
```

Three different ways of writing functions, but they all work in a similar way -
Receive input data, transform data, and then output data.

FIRST-CLASS AND HIGHER-ORDER FUNCTIONS

FIRST-CLASS FUNCTIONS

JavaScript treats functions as first-class citizens.

This means that functions are simply values

Functions are just another “type” of object

Store functions in variables or properties

Pass functions as arguments to OTHER functions

Return functions FROM functions

HIGHER-ORDER FUNCTIONS

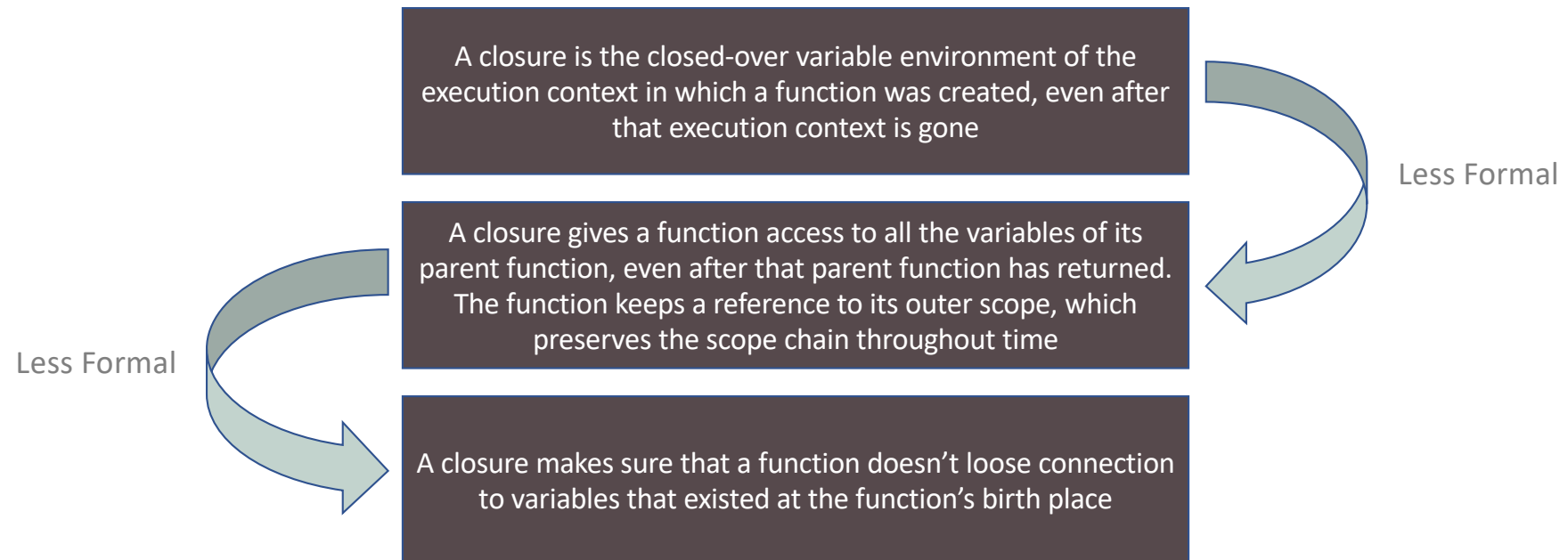
A function that receives another function as an argument, that returns a new function, or both

This is only possible because of first-class functions

Function that receives another function

Function that returns new function

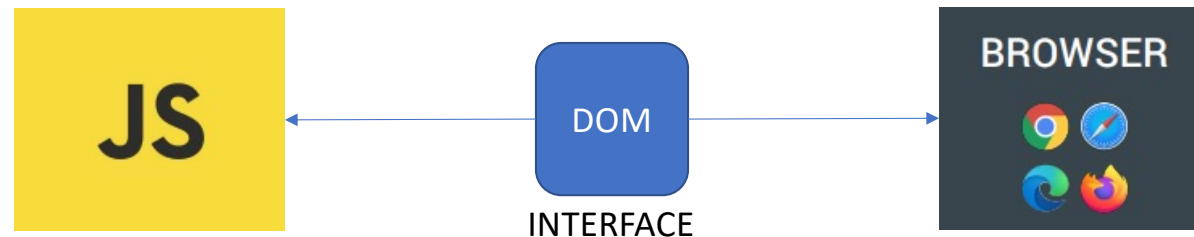
CLOSURES



DOM AND EVENTS

JAVASCRIPT IN THE BROWSER

DOM IN DETAIL

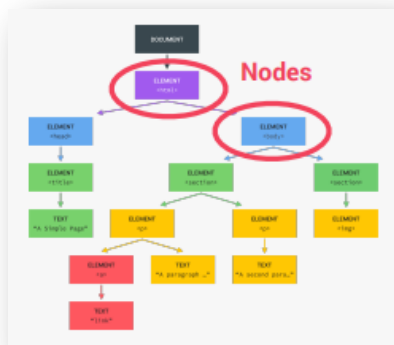


Allows us to make JavaScript interact with the browser

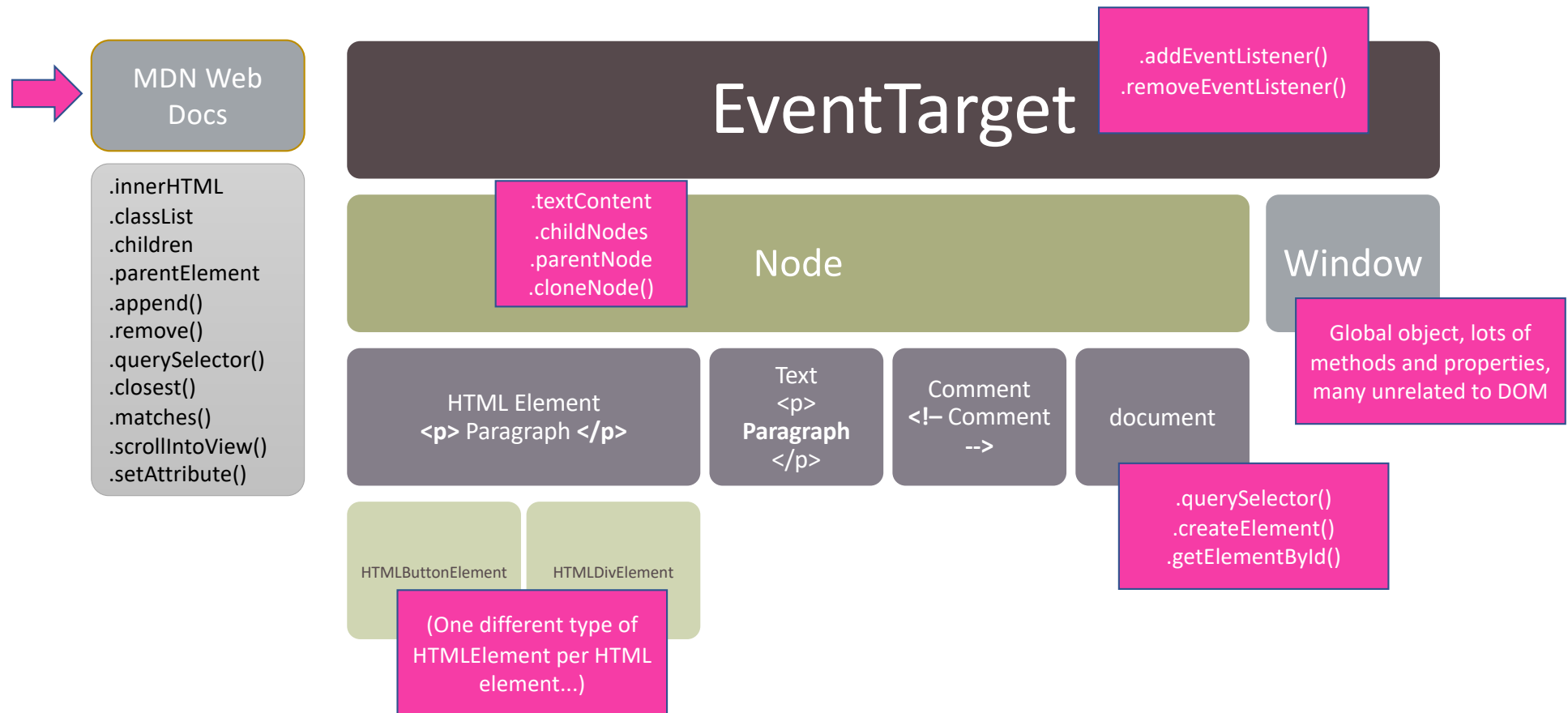
We can write JavaScript to create, modify and delete HTML elements set styles, classes and attributes; and listen and respond to events

DOM tree is generated from an HTML document, which we can then interact with

DOM is a very complex API that contains lots of methods and properties to interact with the DOM tree



HOW THE DOM API IS ORGANIZED BEHIND THE SCENES



JAVASCRIPT NEW FEATURES

ES6 AND MORE

ES6 & MORE – NEW FEATURES

ARROW
FUNCTION

DESTRUCTURING

REST / SPREAD

TEMPLATE
LITERALS

BLOCK SCOPING

MAP/SET

CLASSES

DEFAULT
PARAMETERS

PROMISE API
