Repository of most useful commands and functions in JavaScript

HTML elements

Accessing HTML elements

- `document.getElementById('id')` - returns the element with the specified id
- `document.getElementsByClassName('class')` - returns a collection of all elements with the specified class.
- `document.getElementsByTagName('tag')` - returns a collection of all elements with the specified tag
- `document.querySelector('selector')` - returns the first element that matches the specified selector
- `document.querySelectorAll('selector')` - returns a collection of all elements that match the specified selector

Adding HTML elements

- `document.createElement('div')` - creates a new HTML element
- `document.createElement('ul')` - creates a new unordered list

Modifying HTML elements

- `document.body.appendChild(childElement)` - adds a child element to the end of the parent element
- `element.innerHTML = 'new content'` - changes the inner HTML of an element
- `element.textContent = 'new content'` - changes the text content of an element (ignores HTML tags)

Event listeners and actions

- `element.onclick = function() { ... }` - assigns a function to be executed when the element is clicked
- `element.addEventListener('click', function() { ... })` - assigns a function to be executed when the element is clicked
- `element.addEventListener('mouseover', function() { ... })` - assigns a function to be executed when the mouse pointer is moved onto the element
- `element.addEventListener('mouseout', function() { ... })` - assigns a function to be executed when the mouse pointer is moved out of the element
- `element.addEventListener('change', function() { ... })` - assigns a function to be executed when the value of the element is changed
- `element.addEventListener('input', function() { ... })` - assigns a function to be executed when the value of the element is changed

User input

- `const content = prompt('message')` - displays a dialog box with a message and an input field for the user to enter text

Nested functions

- `function outerFunction() { function innerFunction() { ... } }` - a function can contain another more specific function
- `stopPropagation()` - stops the bubbling of an event to parent elements

Iterations