**CSS**

**CSS3 Features:**

Box-shadow - opacity - Rounded Corners - Attribute Selectors -

New Color(RGBA, HSL, HSLA, Gradient Colors)

=========

**Inheritance:**

Penurunan properti style ke child

=========

**Class bersifat shareable:**

dapat diterapkan pada banyak elemen dengan tipe elemen yang berbeda-beda.

=========

**id:**

tidak bersifat shareable

=========

**[attr~=value]:**

Menargetkan elemen yang menerapkan atribut attr dengan salah satu nilainya adalah value.

**^**

nilainya diawali dengan nilai value.

**$**

nilainya diakhiri dengan value.

**\***

nilainya mengandung value.

===========

**4 KOMBINATOR:**

**Adjacent Sibling Selector (+):**

*~contoh img + p*

menargetkan elemen p pertama setelah image

(harus berada di parent yang sama)

**General Sibling Selector (~)**

*~contoh img + p*

menargetkan elemen p pertama, ketiga setelah image (harus berada di parent yang sama)

**Child Selector (>)**

*~contoh div + p*

menargetkan pada child p dari elemen div

**Descendant Selector ("spasi")**

*~contoh div p*

udah tau gw

===========

**pseudo class (:)**

**pseudo element (::)**

===========

**font: style weight variant size font-family**

===========

**Edge of element :**

merupakan batas dari suatu elemen, Di dalamnya terdapat margin, border, padding dan konten.

===========

**Box Sizing:**

**box-content** ukuran akan berubah jika diberi padding, border

**border-box** ukuran tidak akan berubah walau diberi padding, border

===========

**Padding**

merupakan jarak antara area konten dan border.

===========

**block element:**

* Elemen HTML secara default menambahkan baris baru ketika dibuat.
* Jika tidak diatur lebarnya, lebar dari elemen block akan memenuhi
* lebar dari browser atau elemen yang menaunginya.
* Kita dapat mengatur dimensi dari elemen block.
* Di dalam elemen block, kita dapat menyimpan tag elemen HTML lainnya.

**inline element:**

* Elemen HTML yang secara default tidak menambahkan baris baru ketika dibuat.
* Nilai lebar dan tinggi elemen inline sebesar konten di dalamnya, dan tidak dapat diubah.
* Margin dan padding hanya mempengaruhi elemen secara horizontal, tidak vertikal.

**box-shadow: h v blur spread color**

===========

**FLOW**

**flow normal:**

setiap elemen block ditempatkan di bawah elemen sebelumnya.

**flow tidak normal:**

mengambang, tidak di anggap ada posisinya

===========

**float:**

* dapat membuat elemen berada pada sebelah kanan atau kiri. Saat diterapkan pada elemen inline
* memungkinkan elemen di sekitarnya mengelilingi elemen tersebut (wrap)

jangan lupa menentukan ukuran lebar dari propertinya,

===========

**FLEX**

**flex-grow**: satuan (1)

**flex-basis:** seluruh satuan css, px em, % dll

========================================

**JAVASCRIPT**

let postfix = 5;

console.log(postfix++);

/\* output: 5 \*/

====

let prefix = 5;

console.log(++prefix);

/\* output: 6 \*/

====

let greet = "Hello";

let moreGreet = greet + greet;

console.log(moreGreet);

/\* output: HelloHello \*/

====

const a = 10;

const b = 12;

let isGreater = a > b;

let isLess = a < b;

console.log(isGreater);

console.log(isLess);

/\* output:

false

true

\*/

===========

**Null**

Serupa dengan undefined, namun null perlu diinisialisasikan pada variabel.

null biasa digunakan sebagai nilai sementara pada variabel, tapi sebenarnya

nilai tersebut “tidak ada”.

*contoh:*

let someLaterData = null;

console.log(someLaterData);

/\* output:

null

\*/

===========

**Jika kita mengakses nilai array lebih dari index-nya maka hasilnya akan undefined.**

===========

**OBJECT**

let object = {

key1: "value1",

key2: "value2",

key3: "value3"

};

let user = {

name: {

first: "Harry",

last: "Potter"

},

age: 20,

isMuggle: false,

stuff: ["Magic Wind", "Flying Car", "Owl"]

}

console.log("Hallo, nama saya " + user.name.first + " " + user.name.last);

console.log("Umur saya " + user.age + " tahun")

/\* output

Hallo, nama saya Harry Potter

Umur saya 20 tahun

\*/

===========

**ASSIGNMENT OPERATOR**

let x = 10;

let y = 5

x += y; (x = x + y;)

console.log(x);

/\* output

15

\*/

===========

**COMPERASION OPERATOR**

let a = 10;

let b = 12;

console.log(!(a < 15)); // !(true) -> false

console.log(!(a < 15 && b > 10)); // !(true && true) -> !(true) -> false.

/\* output

false

false

\*/

===========

**alur atau flow proses**

===========

**LOOP**

let myArray = ["Harry", "Ron", "Hermione", "Tom"];

for(let i = 0; i < myArray.length; i++) {

console.log(myArray[i]);

}

for(const arrayItem of myArray) { (LEBIH BARU DAN SIMPEL)

console.log(arrayItem)

}

/\* output

Harry

Ron

Hermione

Tom

\*/

===========

**FUNCTION**

function greeting(name, language) {

    if(language === "English") {

        console.log("Good Morning " + name + "!");

    } else if (language === "French") {

        console.log("Bonjour " + name + "!");

    } else {

        console.log("Selamat Pagi " + name + "!");

    }

}

greeting("Harry", "French");

/\* output

Bonjour Harry!

\*/

function greeting(name, language) {

    if(language === "English") {

        return "Good Morning " + name + "!";

    } else if (language === "French") {

        return "Bonjour " + name + "!";

    } else {

        return "Selamat Pagi " + name + "!";

    }

}

let greetingMessage = greeting("Harry", "French");

console.log(greetingMessage);

/\* output

Bonjour Harry!

\*/

function multiply(a, b) {

    return a \* b;

}

let result = multiply(10, 2)

console.log(result)

/\* output

20

\*/

===========

**MODE STRICT**

Mode Strict adalah mode yang memaksakan penulisan program JavaScript yang baik dan aman.e

Memastikan tidak terjadinya kegagalan menjalankan kode akibat penulisan script yang buruk, dan mencegah timbulnya bug.

hal yang tidak dapat dilakukan jika menggunakan mode Strict:

1. Menggunakan variabel sebelum mendeklarasikannya.
2. Menghapus variabel, function dan argumen.
3. Menggunakan definisi object property yang berulang.
4. Menggunakan definisi parameter pada function yang berulang.
5. Menulis pada property jenis readonly.
6. Menggunakan angka dengan penulisan oktal.
7. Menulis pada property jenis get.
8. Menghapus undelete property seperti pada property jenis prototype.
9. Menggunakan string “eval” dan “argument” sebagai variabel.
10. Menggunakan statement jenis with.
11. Menggunakan future keyword reserve seperti implements, interface, package, private, protected, public, static, yield.

"use strict";

x = 9;       // ini akan dianggap error karena variabel belum dideklarasikan

showAngka();   // ini akan dianggap error karena function belum dideklarasikan

function showAngka() {

    "use strict";

    var x = 9;

    alert(x);

}

===========

**BROWSER OBJECT**

browser dikenal sebagai window object.

**MEMANIPULASI ELEMENT**

Untuk mengubah atau menambah nilai atribut pada elemen, apa  method yang kita gunakan? setAttribute()

someElement.setAttribute("attributName", "attributeValue");

let catImage = document.querySelector("#catImage");

catImage.setAttribute("src", "https://i.ibb.co/55VG7vL/three-cat.jpg")**;**

**MENAMBAHKAN AKSI (EVENT) PADA ELEMENT**

Pada fungsi kita dapat memberikan sebuah argumen yang merupakan object yang berisikan informasi tentang action yang terjadi. Termasuk informasi mengenai target elemen event-nya (event.target). Argumen ini biasanya diberikan nama **“event”**.

catImage.addEventListener("click", function(event) {

document.querySelector("#count").innerText++;

});

**WEB STORAGE**

Untuk menyimpan data gunakan method setItem(), method ini membutuhkan dua argumen yakni key dan nilai yang akan dimasukkan.

Untuk mengakses data yang sudah dimasukkan kita gunakan method getItem() dan gunakan key sebagai argumen methodnya.

**Session Storage**

Data yang simpan akan hilang jika menutup browser atau tab

Untuk menggunakan Session Storage gunakan object sessionStorage,

const cacheKey = "NUMBER";

        if (typeof(Storage) !== "undefined") {

             // pengecekkan apakah data sessionStorage dengan key NUMBER tersedia atau belum

            console.log("Web storage tersedia")

            if (sessionStorage.getItem(cacheKey) === "undefined") {

                // Jika belum maka akan atur dengan nilai awal yakni 0

                sessionStorage.setItem(cacheKey, 0);

            }

            const button = document.querySelector("#button");

            const count = document.querySelector("#count");

            button.addEventListener('click', function(event) {

                let number = sessionStorage.getItem(cacheKey);

                number++; // data akan bertambah setiap #button diklik

                sessionStorage.setItem(cacheKey, number); // dan akan tersimpan di session storage

                count.innerText = sessionStorage.getItem(cacheKey); // baru data dimunculkan di browser

            })

        }

        else {

            alert("Web Storage tidak tersedia")

        }

**Local Storage**

Data yang simpan tidak akan hilang walaupun menutup browser atau tab

Untuk menggunakan Session Storage gunakan object localStorage

const cacheKey = "NUMBER";

        if (typeof(Storage) !== "undefined") {

            console.log("Web storage tersedia")

            // pengecekkan apakah data localStorage dengan key NUMBER tersedia atau belum

            if (localStorage.getItem(cacheKey) === "undefined") {

               // Jika belum maka akan atur dengan nilai awal yakni 0

               localStorage.setItem(cacheKey, 0);

           }

            const button = document.querySelector("#button");

            const clearButton = document.querySelector("#clear");

            const count = document.querySelector("#count");

            button.addEventListener('click', function(event) {

                let number = localStorage.getItem(cacheKey);

                number++;

                localStorage.setItem(cacheKey, number);

                count.innerText = localStorage.getItem(cacheKey);

            });

            clearButton.addEventListener('click', function(event) {

                localStorage.removeItem(cacheKey);

            });

        }

        else {

            alert("")

        }