

Uzoreci	a11	a12
	a21	

```
<div class="grid-co
  <div class="grid-it
    <div class="grid-it
      <div class="grid-it
        <div class="grid-it
```

```

    </div>
    ie(pc) .grid-container{
        grid-template-columns: 1fr 1fr 1fr;
        justify-items: end;
        background-color: blue;
    }

```

```
/* lijevo,centar,desno
align-items: end;
/* gore,sredina,dole
```

```
sno -> start,center,end */
polje -> start,center,end,stretch */ }
```

Važniji globalni atributi: id,class,
lang,title,style

```
var x = 123e-5; let y = "string";
const z = 'string';
// "1"+"2"="12", Number(1)+Number(2)=3
let exp = 2**3 // 2^3 = 8
// strings \
let s = "he said: \"xd\"";
let len = s.length;
let index = s.indexOf("xd"); // .lastIndexOf()
let newString = s.slice(1,2); // [od,do),e
let numToString = (123).toString();
let stringToNum = Number(numToString);
// functions \
function f1(x="default value") {
    console.log(x);
}
let f2 = function(x) {console.log(x);}
let f3 = (x) => console.log(x);
// arrays \
let arr = [1,2,3,5,6,1,7,8];
let last = arr.pop(); // makne i returna zadnji
arr.push(4); // append na kraj
let first = arr.shift(); // makne i returna prvi
arr.unshift(5); // ubaci na pocetak
arr.splice(1,2); // brise 2 elementa od indeksa 1
let arr2 = [6, 7];
let arr3 = arr.concat(arr); // spoji
arr4 = arr3.slice(1,3);
console.log(arr4.includes(6));
arr.sort( function(a,b){return b - a } ); // sort sa komparatorom (desc)
arr.reverse();
arrEven = arr.filter( (x) => x%2==0);
for (let element of arr) {console.log(element);} // elementi
for (let index in arr) {console.log(index);} // indeksi
// classes, objects \
class Person {
    lastName = "Doe";
    age = 50;
    constructor(firstNameValue) {this.firstName = firstNameValue;}
    get lastName() {return this.lastName;}
    set lastName(newLastName) {this.lastName = newLastName}
}
let person1 = new Person("John");
person1.lastName = "Williams";
person1.firstName = "Jake";
for (let value of Object.values(person1)) {console.log(value)}
// try-catch \
try {
    throw "error";
} catch(err) {
    console.log(err);
} finally {
    var x = 2; }
}

let html = document.documentElement;
let body = document.body;
let head = document.head;
let bodyChildrenElements = body.children; // samo html elementi
let bodyChildren = body.childNodes;
// sadrzi i tekst, npr "aa bb" je jedan clan, te svaki \n
let h1 = body.firstElementChild;
// samo html elementi, .firstChild za sve
let script = body.lastElementChild;
// samo html elementi, .lastChild za sve

let p = h1.nextElementSibling;
h1 = p.previousElementSibling;
body = h1.parentElement;
p = document.getElementById("i1");
h1 = document.querySelector(".c2"); // css selektori
p_h1 = document.querySelectorAll(".c2");
p.style.backgroundColor = "Red";
p.innerHTML = "promjena teksta";

Booelan
(undef,null,NaN,0,"")
=false
btn.onclick = () => { //do something};
btn.setAttribute('onclick', 'f()');
f());

setTimeout( () => console.log(3) , 3000 ); // nakon 3s
let promise = new Promise( (resolve, reject) => {
    setTimeout( () => {
        console.log("nakon 3 sekunde...");
        if (false) {
            resolve("dobro izvršen");
        } else {
            reject("lose izvršen");
            // moze i npr reject(new Error("lose izvršen")) }
        },
        3000);
    });
    promise.then(
        function(result) {console.log(result);},
        // ako resolve, onda se ovo izvršava, result = "dobro izvršen"
        function(error) {console.log(error);}
        // ako reject, onda se ovo izvršava, result = "lose izvršen"
    );
    promise.catch(
        function(error) {console.log(error);} // samo ako reject
    )
    promise.catch(
        function(error){console.log(error);}
    ).then(
        function(result){console.log("Resolve:"+result)},
        function(result){console.log("Reject:"+result)}
    ); // catch ce uhvatit error, u then se ce pozvat prva funkcija s result = undefined
    //Fetch\
    let promise2 = fetch("https://web1lab2.azurewebsites.net/products?categoryId=1");
    promise2.then( // obraduje se promise od fetcha
        function(response) {
            if (!response.ok) { throw new Error("Cannot load"); }
            else { return response.json(); } // novo obećanje reponse.json()
        },
        function(error) { throw error; }
    ).then( // obraduje se promise od response.json
        function(response) { console.log("Loaded JSON"); }
    ).catch( // catch hvata gresku u bilo kojem promiseu
        function(error) { console.log(error); }
    )
    //LoadJson\
    async function LoadJSON() { // funkcija se izvodi asinkrono
        let promise = await fetch("https://web/categoryId=1");
        // unutar funkcije, await se izvodi sinkrono (ostatak funkcije ceka)
        if (!promise.ok) { throw new Error ("Cannot load"); }
        else { var jsonContents = await promise.json(); }
        console.log(jsonContents);
    }
    LoadJSON().catch(
        (error) => {console.log(error);}
    )
}

<body>
aa
bb
<h1 class="c2"> Naslov </h1>
<p id="i1" class="c2"> Paragraf </p>
</body>

let newP = document.createElement("p");
newP.innerHTML = "novi paragraf";
body.appendChild(newP);
alert("hey");
let answer = prompt("name?", "default");
let response = confirm("cookies?");
// JSON \
let person = {firstName:"John"
, lastName:"Doe", age:50, eyeColor:"blue"};
let personJSON = JSON.stringify(person);
let personFromJSON = JSON.parse(personJSON);

// local storage \
let ls = window.localStorage;
ls.setItem("item", 2);
let keyAtIndex0 = ls.key(0);
let length = ls.length;
let item = ls.getItem("item");
ls.removeItem("item");
ls.clear();

Module2.js
import sum from "../module1.js"
let result=sum(a,b)

Module1.js
export function sum(a, b){
    return a+b }
```