

Basics ▶

Loops ↻

Ads 🍌

If - Else ↴

```
if ((age >= 14) && (age < 19)) { // logical condition
  status = "Eligible."; // executed if condition is true
} else { // else block is optional
  status = "Not eligible."; // executed if condition is false
}
```

Switch Statement

```
switch (new Date().getDay()) { // input is current day
  case 6: // if (day == 6)
    text = "Saturday";
    break;
  case 0: // if (day == 0)
    text = "Sunday";
    break;
  default: // else...
    text = "Whatever";
}
```

Ads 🍌

Data Types 9L

Output

```
document.getElementById("elementID").innerHTML = "Hello World!";

var age = 18; // number
var name = "Jane"; // string
var name = {first:"Jane", last:"Doe"}; // object
var truth = false; // boolean
var sheets = ["HTML", "CSS", "JS"]; // array
var a; typeof a; // undefined
var a = null; // value null
```

Just praxx, Br
Pranax.com

```
Object
var student = { // object name
  firstName:"Jane", // list of properties and values
  lastName:"Doe",
  age:18,
  height:170,
  fullName: function() { // object function
    return this.firstName + " " + this.lastName;
  }
};
student.age = 19; // setting value
student[age]++; // incrementing
name = student.fullName(); // call object function
```

Numbers and Math Σ

```
var pi = 3.141;
pi.toFixed(0); // returns 3
```

Dates 📅

```
Mon Dec 08 2025 23:00:06 GMT+0100 (Central European Standard Time)
var d = new Date();
1765231206414 milliseconds passed since 1970
Number(d)

Date("2017-06-23"); // date declaration
Date("2017"); // is set to Jan 01
Date("2017-06-23T12:00:00-09:45"); // date - time YYYY-MM-DDTHH:MM:SSZ
Date("June 23 2017"); // long date format
Date("Jun 23 2017 07:45:00 GMT+0100 (Tokyo Time)"); // time zone
```

Get Times

```
var d = new Date();
a = d.getDay(); // getting the weekday

getDate(); // day as a number (1-31)
getDay(); // weekday as a number (0-6)
getFullYear(); // four digit year (yyyy)
getHours(); // hour (0-23)
getMilliseconds(); // milliseconds (0-999)
getMinutes(); // minutes (0-59)
getMonth(); // month (0-11)
getSeconds(); // seconds (0-59)
getTime(); // milliseconds since 1970
```

Setting part of a date

```
var d = new Date();
d.setDate(d.getDate() + 7); // adds a week to a date

setDate(); // day as a number (1-31)
setFullYear(); // year (optionally month and day)
setHours(); // hour (0-23)
setMilliseconds(); // milliseconds (0-999)
setMinutes(); // minutes (0-59)
setMonth(); // month (0-11)
setSeconds(); // seconds (0-59)
setTime(); // milliseconds since 1970
```

Ads 🍌

Global Functions ()

```
eval(); // executes a string as if it was script code
String(23); // return string from number
(23).toString(); // return string from number
Number("23"); // return number from string
```

Strings ⑨

```
var abc = "abcdefghijklmnopqrstuvwxyz";
var esc = 'I don\'t \n know'; // \n new line
var len = abc.length; // string length
abc.indexOf("emo"); // find substring, -1 if doesn't contain
abc.lastIndexOf("emo"); // last occurrence
abc.slice(3, 6); // cuts out "def", negative values count from behind
abc.replace("abc", "123"); // find and replace, takes regular expressions
abc.toUpperCase(); // convert to upper case
abc.toLowerCase(); // convert to lower case
abc.concat(" ", str2); // abc + " " + str2
abc.charAt(2); // character at index: "c"
abc[2]; // unsafe, abc[2] = "c" doesn't work
abc.charCodeAt(2); // character code at index: "c" -> 99
abc.split(","); // splitting a string on commas gives an array
abc.split(""); // splitting on characters
128.toString(16); // number to hex(16), octal (8) or binary (2)
```

Events ①

```
document
a * (b + c) // grouping
person.age // member
person[age] // member
```

Arrays ③

```
var dogs = ["Bulldog", "Beagle", "Labrador"];
var dogs = new Array("Bulldog", "Beagle", "Labrador");

alert(dogs[1]); // access value at index,
dogs[0] = "Bull Terrier"; // change the first item
for (var i = 0; i < dogs.length; i++) { // parsing
  console.log(dogs[i]);
}

Methods
dogs.toString(); // convert to
dogs.join(" * "); // join: "Bull
dogs.pop(); // remove last
dogs.push("Chihuahua"); // add new ele
dogs[dogs.length] = "Chihuahua"; // the same as
dogs.shift(); // remove first
dogs.unshift("Chihuahua"); // add new ele
delete dogs[0]; // change elem
dogs.splice(2, 0, "Pug", "Boxer"); // add element
var animals = dogs.concat(cats, birds); // join two ar
dogs.slice(1, 4); // elements fr
dogs.sort(); // sort string
dogs.reverse(); // sort string
```

JSON j

```
var str = '{"names":[" +
  '{"first":"Hakuna","lastN":"Matata"}', +
  '{"first":"Jane","lastN":"Doe"}', +
  '{"first":"Ala","last":"Jordan"} ]}';
obj = JSON.parse(str); // parse
document.write(obj.names[1].first); // access
```

Just praxx, Br
Pranax.com

```
var myObj = { "name":"Jane", "age":18, "city":"Chicago" }; // create object
var myJSON = JSON.stringify(myObj); // stringify
window.location = "demo.php?x=" + myJSON; // send to php

Storing and retrieving
myObj = { "name":"Jane", "age":18, "city":"Chicago" };
myJSON = JSON.stringify(myObj); // storing data
localStorage.setItem("testJSON", myJSON);
text = localStorage.getItem("testJSON"); // retrieving data
obj = JSON.parse(text);
document.write(obj.name);
```

Useful Links ↗

Regular Expressions \n

```
var a = str.search(/CheatSheet/1);
```

Modifiers

```
i perform case-insensitive matching
g perform a global match
m perform multiline matching
```

Patterns

```
\ Escape character
\d find a digit
\s find a whitespace character
\b find match at beginning or end of a word
```

Errors ⚠

```
try { // block of code to tr
  undefinedFunction();
} catch(err) { // block to handle err
  console.log(err.message);
}

Throw error
throw "My error message"; // throw a text
```

Promises P

```
function sum (a, b) {
  return Promise(function (resolve, reject) {
    setTimeout(function () {
      if (typeof a !== "number" || typeof b !== "number") {
        return reject(new TypeError("Inputs must be numbers"));
      }
      resolve(a + b);
    }, 1000);
  });
}

var myPromise = sum(10, 5);
myPromise.then(function (result) {
  document.write(" 10 + 5 = ", result);
});
```

return another promise

[JS cleaner](#)[Obfuscator](#)[Can I use?](#)[Node.js](#)[jQuery](#)[RegEx tester](#)

| because of the error
er is called instead, after another second
// => Please provide two numbers to sum.

```
console.error(err);  
});
```

States

pending, fulfilled, rejected

Properties

Promise.length, Promise.prototype

Methods

Promise.all(iterable), Promise.race(iterable), Promise.reject(reason), Promise.resolve(value)