devtown

JavaScript Basics
CHEAT SHEET

Console.log

```
console.log(“JavaScript Cheat Sheet”);
```

Variables & Constant

Variables

```
var variable_1; // global scope
let variable_2; // block scope
```

Constants

```
const constantVariable; // block scope
```

Data Types

Number

```
let num = 3;
let floatData = 3.3;
```

Strings

```
const stringVaribale = “hello”;
```

Arrays

```
let fruit = ["Banana", "Apple", "Orange"];
```

Boolean

```
const isRaining = true;
const isAuth = false;
```

Objects

```
let user = {
  fullname: “Bucky Barnes”,
  age: 40,
  gender: “male”
}
```

Conditional Statements

if statement

```
if (isRaining){
 wearRainCoat();
}
```

if else statement

```
if (isRaining){
 wearRainCoat();
} else {
 doNotWearRainCoat();
}
```

else if statement

```
if (isRaining){
 wearRainCoat();
} else if (isRainingHeavy){
  doNotGoOut();
} else {
 doNotWearRainCoat();
}
```

Array & it's methods

An array

```
let ourArray = [“🔥”, “🎂”, “🍔”, “🍕”];
```

Access array elements

```
ourArray[0]; //🔥
```

Push

```
ourArray.push(“🥚”);
// output: [“🔥”, “🎂”, “🍔”, “🍕”,“🥚”]
```

Shift

```
ourArray.shift()🥚;
// output: [“🎂”, “🍔”, “🍕”, “🥚”] --> “🔥” removed
```

UnShift

```
ourArray.unshift(“🍺”);
// output: [“🍺”, “🔥”, “🎂”, “🍔”, “🍕”, “🥚”]
```

Combine two array - concat()

```
const combine= [“🍔”, “🍕”, “🥚”].concat([“🍺”, “🔥”, “🎂”]);
// Output: [“🍔”, “🍕”, “🥚”,“🍺”, “🔥”, “🎂”]
```

forEach

```
ourArray.forEach((elm) => console.log(elm));
```

Function

sample function

```
function sayHi() {
    console.log("Welcome to this Earth!");
}
```

call a function

```
sayHi();
```

devtown

JavaScript Basics

CHEAT SHEET 02

Arrow function

```
const sayHiAgain = () =>
console.log("Welcome to this Earth!");
```

call an arrow function

```
sayHiAgain();
```

Loops

for loop

```
for (let i = 1; i <= 5; i++) {
    console.log(`This is for loop.`);
}
```

while loop

```
while (i <= 3) {
    console.log(`This is while loop.`);
    i += 1;
}
```

do while loop

```
do {
   console.log(`This is do while loop.`);
  i++;
}
while (i < 5);
```

Scope

Global Scope

The greet variable is accessible everywhere.

```
let greet = `hello`;
function sayHello () {
    console.log(greet);
}
sayHello(); // hello
```

Local Scope

The greet variable is accessible only inside sayhello function.

```
function sayHello () {
    let greet = `hello`;
    console.log(greet);
}
sayHello(); // hello
```

Objects & it's methods

The Object

```
let user = {
  fullname: “Thomas shelby”,
  place: “birmingham”
}
```

Access properties

```
console.log(user.fullname);
// Thomas Shelby
console.log(user.place);
// birmingham
```

entries

```
console.log(Object.entries(user));
// output
// [ ["fullname", "Thomas shelby"],
//    ["place", "birmingham"]]
```

freeze

```
Object.freeze(user);
```

keys

```
Object.keys(user);
// output : [“fullname”, “place”];
```

values

```
Object.values(user);
// output : [“Thomas shelby”, “birmingham”];
```

Assignment Operators

=	Assignment operator
+=	Addition assignment
-=	Subtraction Assignment
*=	Multiplication Assignment
/=	Division Assignment
%=	Remainder Assignment
**=	Exponentiation Assignment

Arithmetic Operators

+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Remainder
**	Exponentiation (Power)
++	Increment (increments by 1)
--	Decrement (decrements by 1)

Comparison Operators

==	Equal to
!=	Not equal to
===	Strict equal to
!==	Strict not equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to

Bitwise Operators

&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR
~	Bitwise NOT
<<	Left shift
>>	Sign-propagating right shift
>>>	Zero-fill right shift

Logical Operators

&&	Logical AND
||	Logical OR
!	Logical NOT