

# The Ultimate Cheat Sheets Guide for Web Developers

BY HASAN TOOR

# 1. HTML CHEAT SHEET

## HTML CHEAT SHEET

**<!DOCTYPE >** Every HTML document must begin with this.

**< html >** → Represent the root element . All other elements must be descendants of this element.

**< head >** → Head tag contains metadata about the document like title , style sheets and scripts.

**< title >** → Defines a title for the document and very important for SEO.

By @Hasanstack

**< body >** → Represents the content of an HTML document.

### CONTENT SECTIONING

<b>&lt; header &gt;</b>	The < header > element represents a container for introductory content or a set of navigational links .
<b>&lt; h1 &gt; to &lt; h6 &gt;</b>	represents six levels of headings . < h1 > is the highest one and < h6 > is the lowest .
<b>&lt; main &gt;</b>	The < main > tag specifies the main content of a document .
<b>&lt; nav &gt;</b>	The < nav > tag defines a set of navigation links , directing users within document or to another page .
<b>&lt; section &gt;</b>	The < section > element represents a standalone section .
<b>&lt; aside &gt;</b>	It represents a portion of a document whose content is indirectly related to the document's main content .
<b>&lt; address &gt;</b>	The < address > tag defines the contact information .
<b>&lt; footer &gt;</b>	The < footer > tag defines a footer for a document or section . It mainly contains information about the author of the section , copyright data or links to related documents .

### TEXT SECTIONING

<b>&lt; div &gt;</b>	The < div > tag is used as a container for HTML elements .
<b>&lt; dl &gt; &amp; &lt; dd &gt;</b>	The < dl > tag defines a description list . The < dd > tag is used to describe a termname in a description list.
<b>&lt; ul &gt;</b>	unordered list of items , typically rendered as a bulleted list .
<b>&lt; ol &gt;</b>	represents an ordered list of items , typically rendered as a numbered list.
<b>&lt; li &gt;</b>	The HTML < li > element is used to represent an item in a list .
<b>&lt; p &gt;</b>	The HTML < p > element represents a paragraph .

# 2. HTML CHEAT SHEET

## HTML Cheat Sheet

BY Hasantoxr

### Document Outline

<!DOCTYPE>  
<html>  
<head>  
<body>

Version of (X)HTML  
HTML document  
Page information  
Page contents

### Document Structure

<h[1-6]>  
<div>  
<span>  
<p>  
<br />  
<hr />

Heading  
Page section  
Inline section  
Paragraph  
Line break  
Horizontal rule

### Form Events

<form>  
<fieldset>  
<legend>  
<label>  
<input />  
<select>  
<optgroup>  
<option>  
<textarea>  
<button>

Form  
Collection of fields  
Form legend  
Input label  
Form input  
Drop-down box  
Group of options  
Drop-down options  
Large text input  
Button

### Form Events

onBlur  
onChange  
onFocus  
onReset  
onSelect  
onSubmit

### Lists

<ol>  
<ul>  
<li>  
<dl>  
<dt>  
<dd>

Ordered list  
Unordered list  
List item  
Definition list  
Definition term  
Term description

### Text Markup

<strong>  
<em>  
<blockquote>  
<a>  
<abbr>  
<acronym>  
<address>  
<pre>  
<dfn>  
<code>  
<cite>  
<del>  
<ins>  
<sub>  
<sup>  
<bdo>

Strong emphasis  
Emphasis  
Long quotation  
Short quotation  
Abbreviation  
Acronym  
Address  
Pre-formatted text  
Definition  
Code  
Citation  
Deleted text  
Inserted text  
Subscript  
Superscript  
Text direction

### Tables

<table>  
<caption>  
<thead>  
<tbody>  
<tfoot>  
<colgroup>  
<col />  
<tr>  
<th>  
<td>

Table  
Caption  
Table header  
Table body  
Table footer  
Column group  
Column  
Table row  
Header cell  
Table cell

# 3. HTML INPUT TYPES CHEAT SHEET

HTML

## HTML Input types Cheat Sheet

BY @Hasantoxr

### Type Text

`<input type="text">`  
defines a single-line text input field:

First name:  
Hasan  
Last name:  
Stack

### Input Type Password

`<input type="password">`  
defines a password field:

Username:  
hasan@gmail.com  
Password:  
.....

### Input Type Radio

`<input type="radio">`  
defines a radio button.

- HTML
- CSS
- JavaScript

### Input Type Checkbox

`<input type="checkbox">`  
defines a checkbox.

- I have a bike
- I have a car
- I have a boat

### Input Type Button

`<input type="button">`  
defines a button:

Click Me!

### Input Type Submit

`<input type="submit">`  
defines a button for submitting  
form data to a form-handler.

### Input Type Date

The `<input type="date">`  
is used for input fields  
that should contain a date.

### Type Datetime-local

The `<input type="datetime-local">`  
specifies a date and  
time input field, with no time zone.

### Input Type Email

The `<input type="email">`  
is used for input fields that should  
contain an e-mail address.

### Input Type Search

The `<input type="search">`  
is used for search fields  
(a search field behaves like a regular text field).

### Input Type Image

The `<input type="image">`  
defines an image  
as a submit button.

### Input Type Week

The `<input type="week">`  
allows the user  
to select a week and year.

### Input Type Time

The `<input type="time">`  
allows the user  
to select a time (no time zone).

### Input Type Url

The `<input type="url">`  
is used for input fields that  
should contain a URL address.

### Input Type Month

The `<input type="month">`  
allows the user to select a month  
and year.

### Input Type Tel

The `<input type="tel">`  
is used for input fields that  
should contain a telephone number.

### Input Type Search

The `<input type="search">`  
is used for search fields  
(a search field behaves like a regular text field).

# 4. CSS BASIC CHEAT SHEET

## CSS BASIC CHEATSHEET

BY Hasantoxr

### Selectors

*	all elements
div	all div tags
div,p	all divs and paragraphs
div p	paragraphs inside divs
div > p	all p tags, one level deep in div
div + p	p tags immediately after div
div ~ p	p tags preceded by div
.classname	all elements with class
#idname	element with ID
div.classname	divs with certain classname
div#idname	div with certain ID
#idname	all elements inside #idname

### Pseudo Classes

a:link	link in normal state
a:active	link in clicked state
a:hover	link with mouse over it
a:visited	visited link
p::after{content:"yo";}	add content after p
p::before	add content before p
input:checked	checked inputs
input:disabled	disabled inputs
input:enabled	enabled inputs
input:focus	input has focus
input:in-range	value in range

### Box Model

#### Margin

#### Border

#### Padding

#### Content

### Fonts & Text

font-family:	<font>, <fontN>
font-size:	<size>
letter-spacing:	<size>
line-height:	<number>
font-weight:	bold normal
font-style:	italic normal
text-decoration:	underline none
text-align:	left right center justify
text-transform:	capitalize uppercase lowercase

### Colors

color name

rgb(x%,y%,z%)

rgb(x,y,z)

#rrggbba

### Positions

Display  
Position  
Top  
Right  
Bottom  
Left  
Float

Clear  
Z-index  
direction  
unicode-bidi  
overflow  
clip  
visibility

# 5. CSS SELECTORS CHEAT SHEET

## CSS Selectors Cheatsheet

BY Hasantoxr

class  
.intro

Selects all elements with class="intro"

#id

#firstname Selects the element with id="firstname"

\*

\* Selects all elements

element

p Selects all <p> elements

element.class

p.intro Selects all <p> elements with class="intro"

element,element

div, p Selects all <div> elements and all <p> elements

element element

div p Selects all <p> elements inside <div> elements

element>element

div > p Selects all <p> elements where the parent is a <div> element

element+element

div + p Selects the first <p> element that is placed immediately after <div> elements

element1~element2

p ~ ul Selects every <ul> element that is preceded by a <p> element

[attribute]

[target] Selects all elements with a target attribute

[attribute=value]

[target=\_blank] Selects all elements with target="\_blank"

[attribute~=value]

[title~=flower] Selects all elements with a title attribute containing the word "flower"

[attribute^=value]

a[href^="https"] Selects every <a> element whose href attribute value begins with "https"

[attribute\$=value]

a[href\$=".pdf"] Selects every <a> element whose href attribute value ends with ".pdf"

:active

a:active Selects the active link

::after

p::after Insert something after the content of each <p> element

::before

p::before Insert something before the content of each <p> element

:checked

input:checked Selects every checked <input> element

:default

input:default Selects the default <input> element

:first-child

p:first-child Selects every <p> element that is the first child of its parent

:hover

a:hover Selects links on mouse over

# 6. CSS UNITS CHEAT SHEET

## CSS Units Cheat Sheet

BY Hasantoxr

**px** → Absolute pixel value

**rem** ↗ Relative to the font size of the root element

**em** ↗ Relative to the font size of the element

**vh** ↗ Relative to 1% of the viewport's height

**vw** ↗ Relative to 1% of the viewport's width

**%** ↗ Relative to 1% of the parent element

**vmin** ↗ Relative to the viewport's smaller dimension,  $1vmin = \max(1vh, 1vw)$

**vmax** ↗ Relative to the viewport's larger dimension,  $1vmax = \max(1vh, 1vw)$

**ch** ↗ Relative to the width of the glyph "O" of the element's font

# 7. CSS CENTER A DIV CHEAT SHEET

## CSS CENTER ALIGN CHEATSHEET

BY Hasantoxr

### TRANSFORM

```
.center {  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  transform: translate(-50%, -50%);  
}
```

### FLEX

```
.parent {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```

### NEGATIVE MARGIN

```
.center {  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  margin: -50px -100px;  
}
```

### GRID

```
.parent {  
  display: grid;  
}  
.center {  
  margin: auto;  
}
```

### FIXED PADDING

```
.parent {  
  height: 600px;  
  padding: 200px 0;  
}  
.center {  
  margin: 0 auto;  
  height: 100%;  
}
```

### TABLE CELL

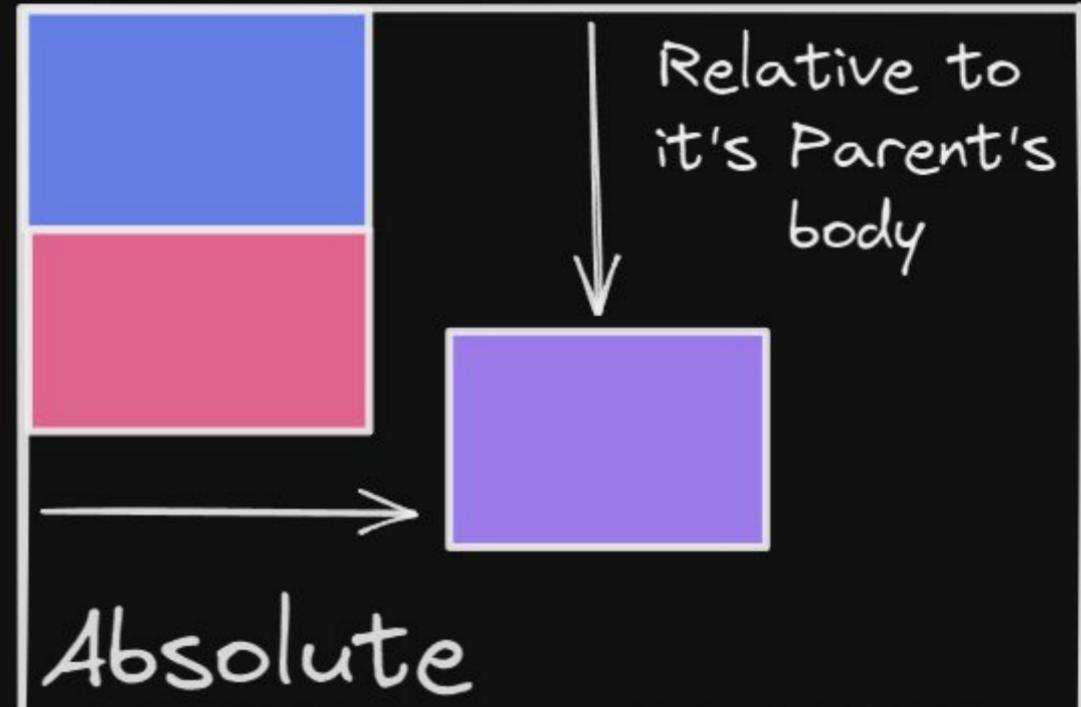
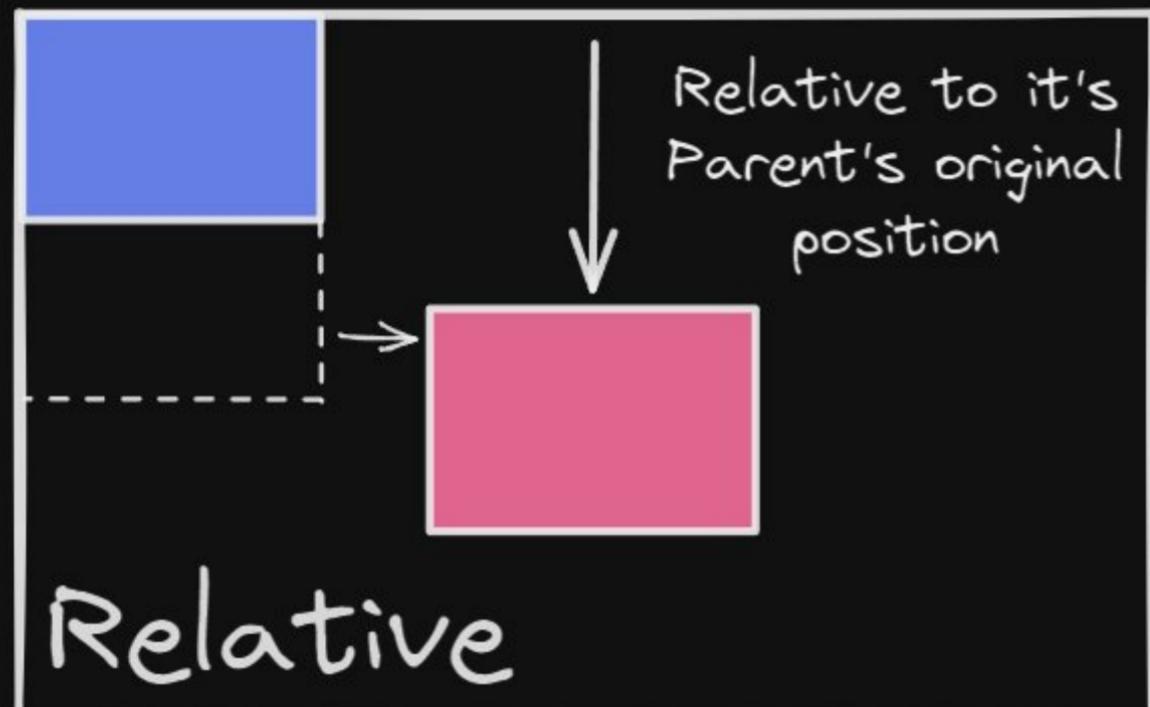
```
.parent {  
  display: table-cell;  
  vertical-align: middle;  
}  
.center{  
  margin: auto;  
}
```

### HORIZONTAL CENTERING

```
.parent {  
  text-align: center;  
}
```

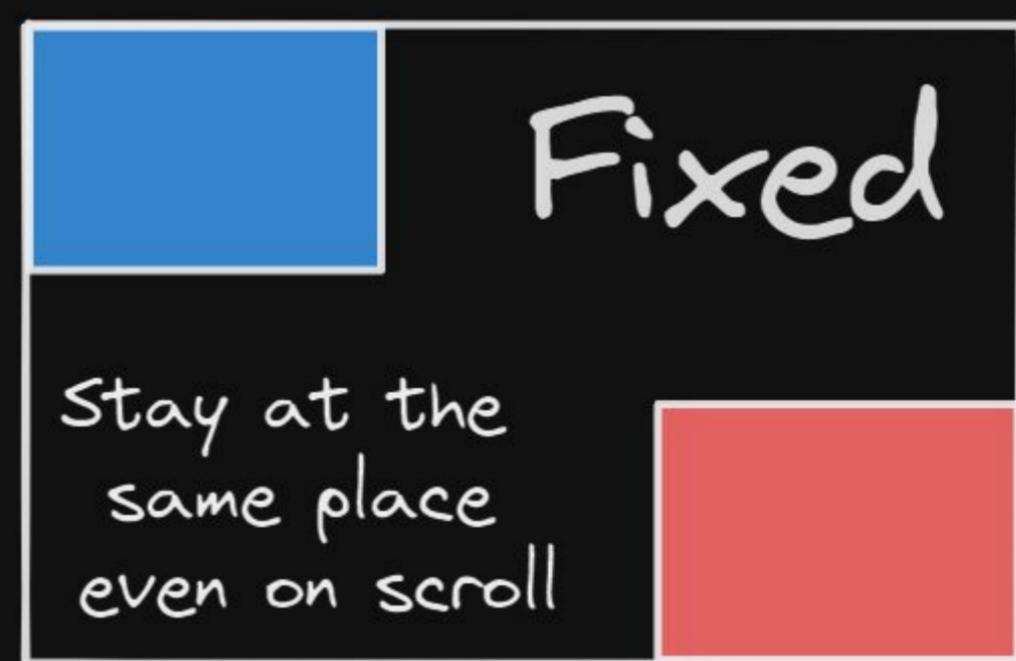
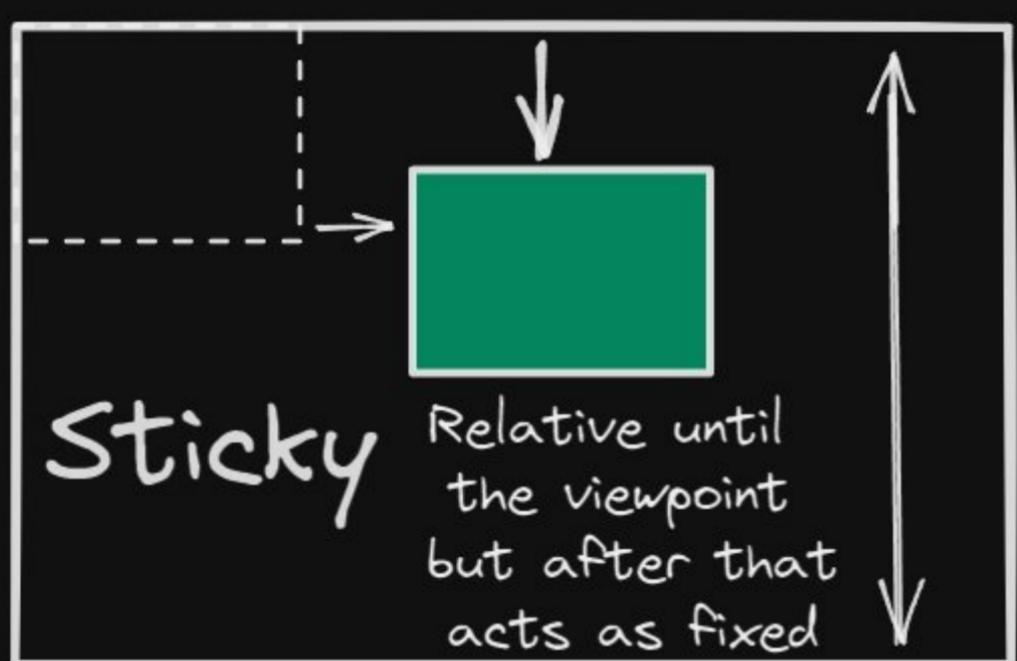
```
.center {  
  margin: 0 auto;  
}
```

# CSS POSITIONING CHEAT SHEET



## CSS Positioning Cheat Sheet

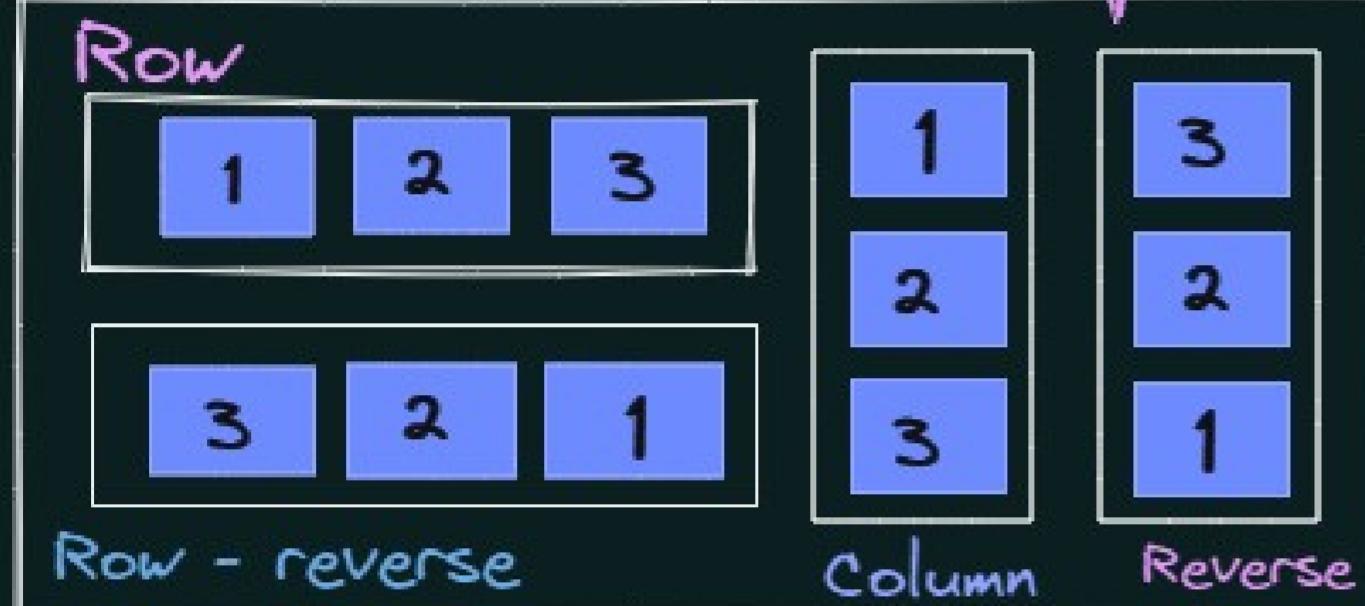
BY @Hasantoxr



# 8. CSS FLEXBOX CHEAT SHEET

## CSS FLEXBOX CHEATSHEET

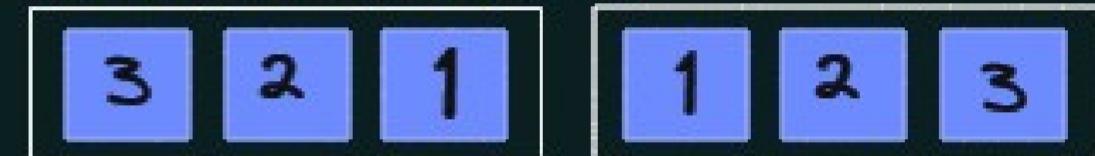
### Flex Container



### Flex Items



### Flex Items Order



### Justify Content

Flex-start



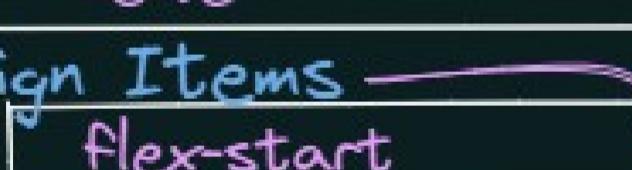
Space - between



Flex-end



Center



Space - around

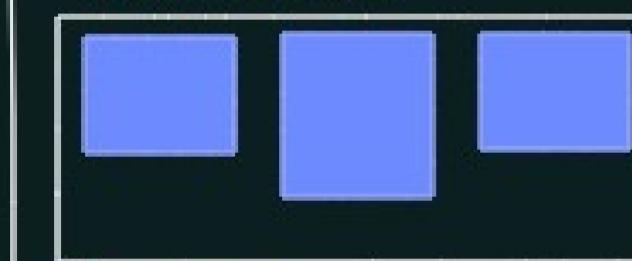


Space - evenly

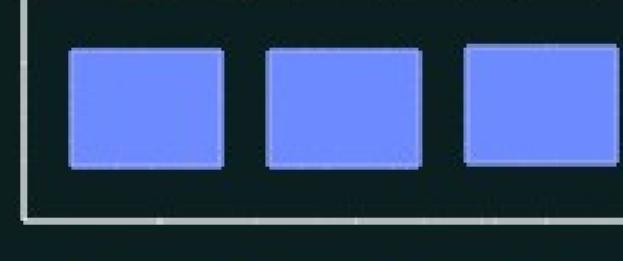


### Align Items

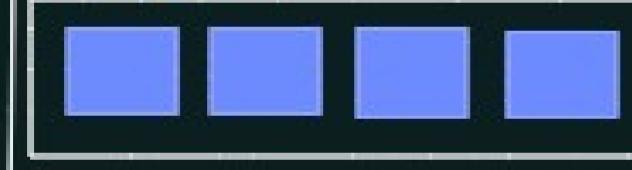
flex-start



center



flex-end



baseline

### Align-Content

flex-start



flex-end



normal

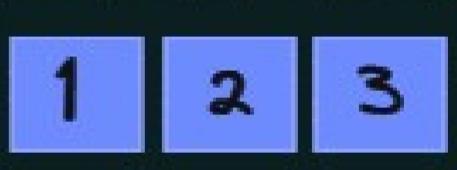
### Flex Grow

how each child grow in remaining space



### Flex Shrink

set amount of flex item shrink relative to others



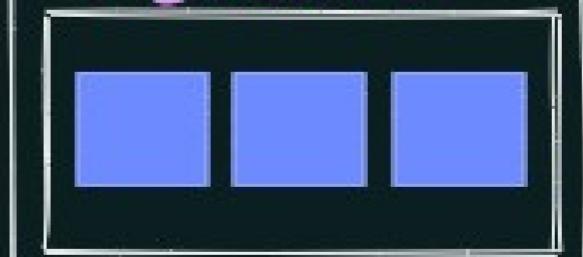
### Flex Basis

default size of a flex item  
First item 20%  
Second item 40%

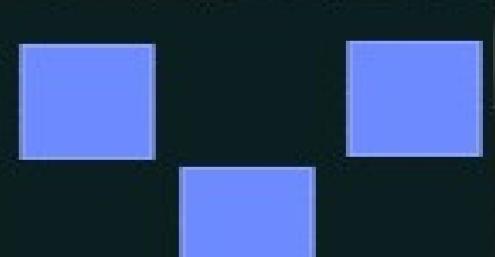


Align Self allows the default alignment

Flex-start



flex-end



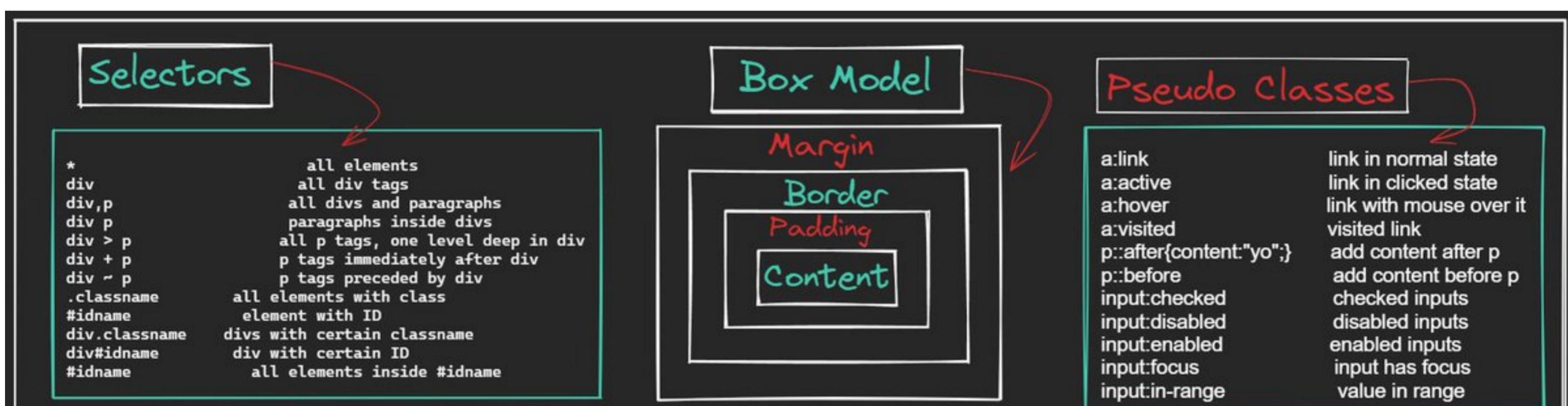
center



stretch

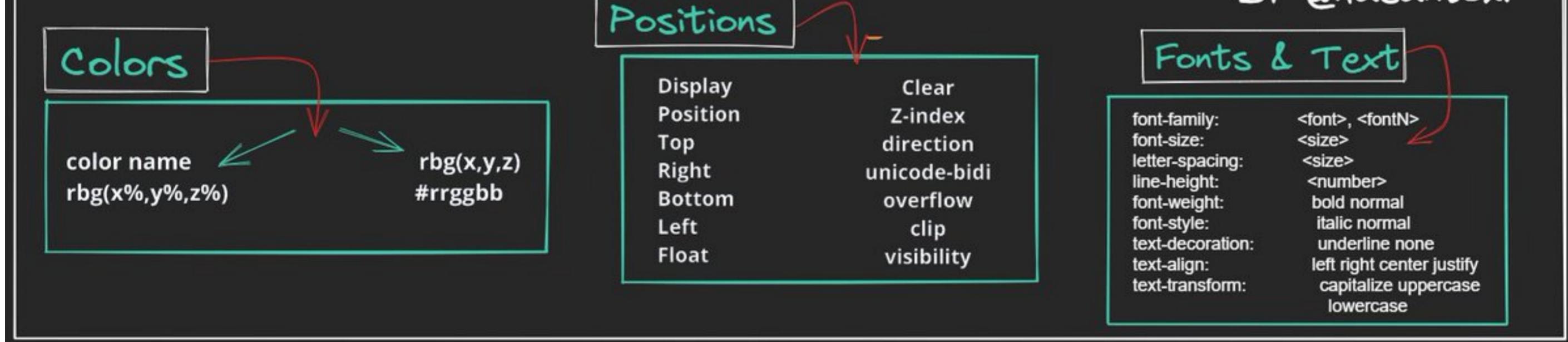


# CSS BASIC CHEAT SHEET



## CSS BASIC CHEATSHEET

BY @Hasantoxr



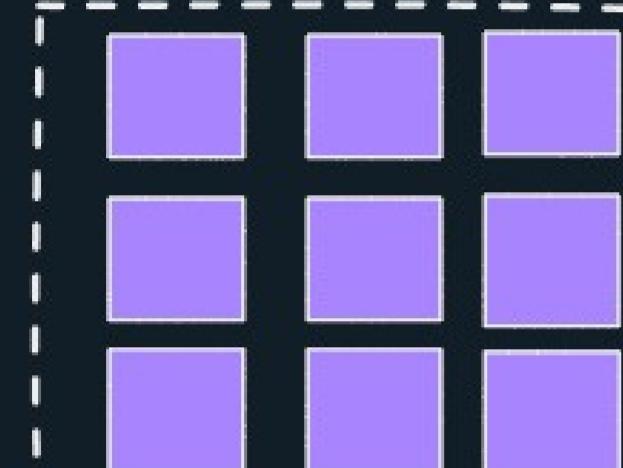
# 9. CSS GRID CHEAT SHEET

## CSS GRID CHEAT SHEET

BY @Hasantoxr

### Grid Architecture

Columns



Grid Container

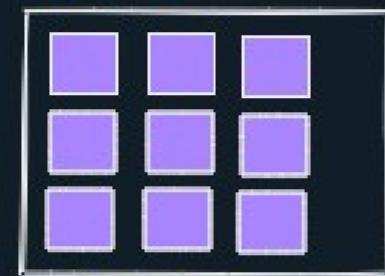
Main Axis

Cross Axis

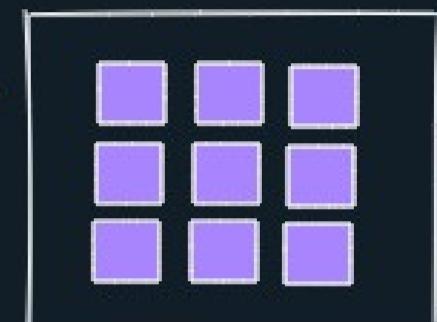
Rows

### Justify-items

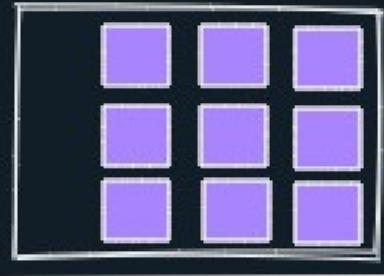
Start



Center

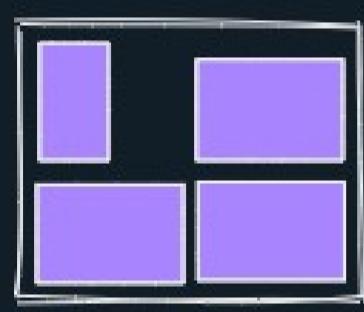


Stretch

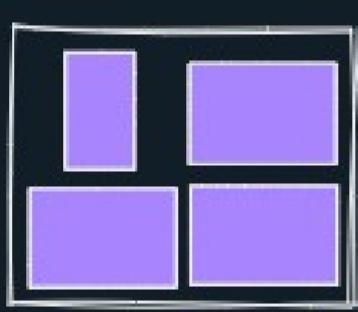


End

### Justify-self



Start



Center

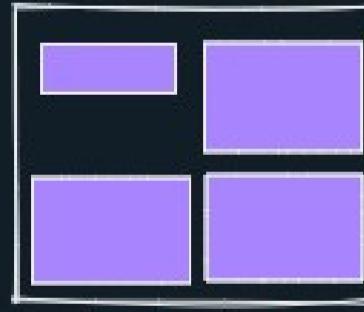


End



Stretch

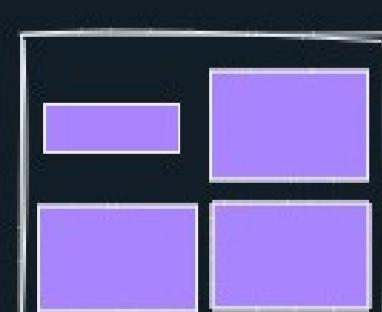
### Align-self



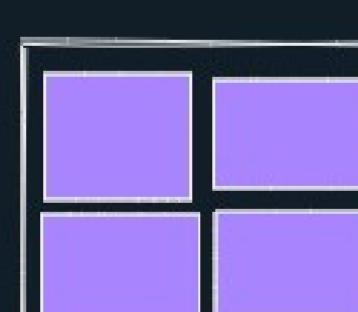
Start



End



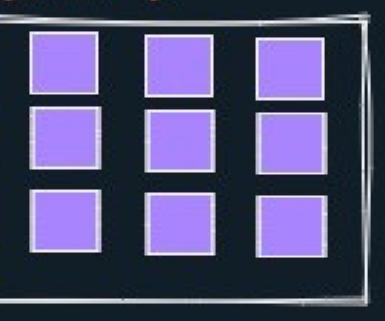
Center



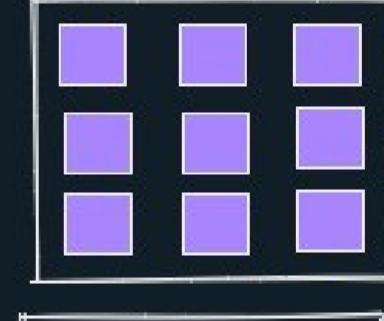
Stretch

### Align-items

Start



Center



Stretch



End

### Justify-content

Start



Start



End

Stretch

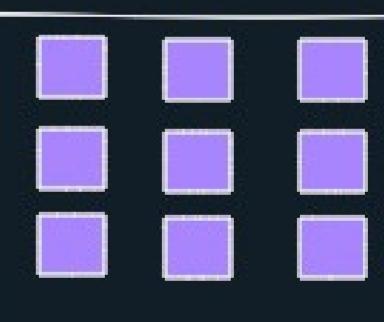


Stretch

Center



Space-between



Space-around

Space-evenly

# 10. JAVASCRIPT BASIC CHEAT SHEET

## JAVASCRIPT BASIC CHEATSHEET

### Variables

To declare a variable in JavaScript

- `var` is used in pre-ES6 versions of JavaScript.
- `let` is the preferred way to declare a variable when it can be reassigned.
- `const` is the preferred way to declare a variable with a constant value.

Example -

```
var age;  
let weight;  
const numberOfFingers = 20;
```

### LOOP

#### While Loop

```
while (condition) {  
  // code block to be executed  
}  
  
let i = 0;  
while (i < 5) {  
  console.log(i);  
  i++;  
}
```

#### For Loop

```
for (let i = 0; i < 4; i += 1) {  
  console.log(i);  
};  
// Output: 0, 1, 2, 3
```

#### Do...While Statement

```
x = 0  
i = 0  
do {  
  x = x + i;  
  console.log(x)  
  i++;  
} while (i < 5);  
// Prints: 0 1 3 6 10
```

# JS

### Conditionals

Logical Operator `||` OR  
`true || false;` // true  
`10 > 5 || 10 > 20;` // true  
`false || false;` // false  
`10 > 100 || 10 > 20;` // false

Logical Operator `! NOT`  
`let lateToWork = true;`  
`let oppositeValue = !lateToWork;`  
`console.log(oppositeValue);`  
// Prints: false

Logical Operator `&&` AND  
`true && true;` // true  
`1 > 2 && 2 > 1;` // false  
`true && false;` // false  
`4 === 4 && 3 > 1;` // true

Comparison Operators  
`1 > 3` // false  
`3 > 1` // true  
`250 >= 250` // true  
`1 === 1` // true  
`1 === 2` // false  
`1 === '1'` // false

### Arithematic Operators

// Addition  
`5 + 5`  
// Subtraction  
`10 - 5`  
// Multiplication  
`5 * 10`  
// Division  
`10 / 5`  
// Modulo  
`10 % 5`

### Functions

```
function greet () {  
  alert ('Hello, ${ name } ' );  
}  
greet ();
```

#### Ternary Operator

```
let price = 10.5;  
let day = "Monday";
```

```
day === "Monday" ? price -= 1.5 : price += 1.5;
```

#### Arrow Function

```
let sum = (x, y) => x + y;
```

```
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
```

### Arrays

// An array containing numbers  
`const numberArray = [0, 1, 2, 3];`

// An array containing different data types  
`const mixedArray = [1, 'chicken', false];`

### Objects

An object is a built-in data type for storing key-value pairs.

```
const student = {  
  name: 'Sheldon',  
  score: 100,  
  grade: 'A',  
}
```

# 11. JAVASCRIPT STRING CHEAT SHEET

## Javascript String Cheatsheet

BY Hasantoxr

### String declaration

```
Const name = "Rajvir";  
Const city = "Sangrur";  
Const msg = `${name} is  
from ${city}`;
```

### String functions

```
msg.indexOf(city); //15  
msg.lastIndexOf(name); //0  
city.charAt(2); // 'y'  
city[3]; // 'y'
```

### String functions

```
city.replace("S", "R");  
city.toUpperCase();  
name.toLowerCase();  
name.concat(" is good")
```

### String functions

```
name.slice(3, 6); //'vir'  
city.split("y"); // [ 'Sa' 'grur']  
name.length; //6
```

# 12. JAVASCRIPT DOM CHEAT SHEET

## Javascript DOM Cheatsheet

BY Hasantoxr

```
< div id = " app " class = " hero dark " >
  < h1 > Hasan is a Developer < / h1 >
< / div >
```

```
// single element defined by id
const app = document.getElementById ( ' app ' )

// multiple elements ( arrays ) defined by class name
const hero = document.getElementsByClassName ( ' hero ' )

// multiple elements based on html tag
const h1 = document.getElementsByTagName ( ' h1 ' )

// first element based on selector
const hero = document.querySelector ( ' . hero ' )

// multiple elements based on selector
const heroes = document.querySelectorAll ( ' . hero ' )
```

```
// create html element of tag < p >
let para = document.createElement ( ' p ' )

// create text node
let text = document.createTextNode ( ' India ' )

// add text node to the para element
para.appendChild ( text )

// insert h2 before h1
app.insertBefore ( h2 , h1 )
< p > India < / p >
```

```
h1.insertAdjacentHTML ( ' beforebegin ' , ' < span > cool < / span > ' )
  // beforebegin = > placed before h1 as a sibling
  // afterbegin = > placed inside h1 as a first child
  // beforeend = > placed inside h1 as a last child
  // afterend = > placed after h1 as a sibling
```

app.classList.remove ( ' dark ' )	// remove class
app.classList.add ( ' light ' )	// add class
app.classList.toggle ( ' visible ' )	// toggle class
app.classList.contains ( ' app ' )	// true if app present
app.childNodes	// retrieve all child nodes
app.parentNode	// return parent node

# 13. JAVASCRIPT ARRAY CHEAT SHEET

JS

## JavaScript Array Methods

BY Hasantoxr

### .concat()

Joins two or more arrays, and returns a copy of the joined arrays

### .every()

Checks if every element in an array pass a test

### .filter()

Creates a new array with every element in an array that pass a test

### .findIndex()

Returns the index of the first element in an array that pass a test

### .forEach()

Calls a function for each array element

### .includes()

Check if an array contains the specified element

### .indexOf()

Search the array for an element and returns its position

### .join()

Joins all elements of an array into a string

### .unshift()

This method adds new elements to the beginning of an array. This method overwrites the original array.

### .toString()

Converts an array to a string, and returns the result

### .map()

Creates a new array with the result of calling a function for each array element

### .pop()

Removes the last element of an array, and returns that element

### .push()

Adds new elements to the end of an array, and returns the new length

### .reduce()

Reduce the values of an array to a single value (going left-to-right)

### .reverse()

Reverses the order of the elements in an array

### .shift()

Removes the first element of an array, and returns that element

### .slice()

Selects a part of an array, and returns the new array

### .some()

Checks if any of the elements in an array pass a test

### .sort()

Sorts the elements of an array

### .splice()

Adds/Removes elements from an array

# 14. FRONTEND ROADMAP CHEAT SHEET

## Front End Dev Roadmap BY @Hasan

### HTML

- Elements + Attributes
- Forms + Input Types
- Imports + META Tags
- Lists & Anchor
- Tables
- Events

- Web Storage
- Images & Multimedia
- Frame
- Canvas & WebGL
- Semantic HTML

### CSS

- Selectors (Basic)
- Pseudo Selectors
- Specify & Inheritance
- Box Model
- Typography
- Positioning, Units, Display

- Layouts : Flex, Grid
- Shadows
- Colors + Gradients
- Transforms & Transitions
- Animations
- Media Queries
- CSS Variables

### JS

- Variables & Primitives
- Scopes & Hoisting
- Operators & type Conversions
- Closures
- Objects + methods
- Arrays + Methods
- Functions & arrow functions
- Timeout & interval

- Modules
- Classes & Proxies
- ES6 + Syntaxes
- Prototypes
- Destructuring & Spread
- "this" keyword
- Callbacks + Promise
- Async Await

# 15. JAVASCRIPT ARRAY CHEAT SHEET

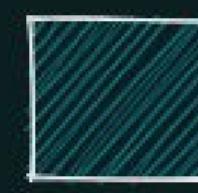
## Array Methods Cheatsheet

JS

BY Hasantoxr



.map(  $\square \rightarrow \circ$  )  $\rightarrow$



.filter(  $\square$  )  $\rightarrow$



.find(  $\square$  )  $\rightarrow$



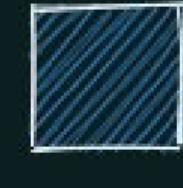
.findIndex(  $\square$  )  $\rightarrow$  3



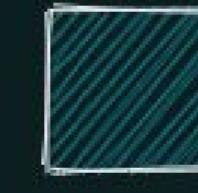
.fill( 1,  $\circ$  )  $\rightarrow$



.copyWithin(1,  $\circ$  )  $\rightarrow$



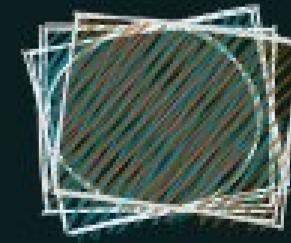
.some(  $\square$  )  $\rightarrow$  true



.every(  $\square$  )  $\rightarrow$  false



.reduce( acc + curr )  $\rightarrow$

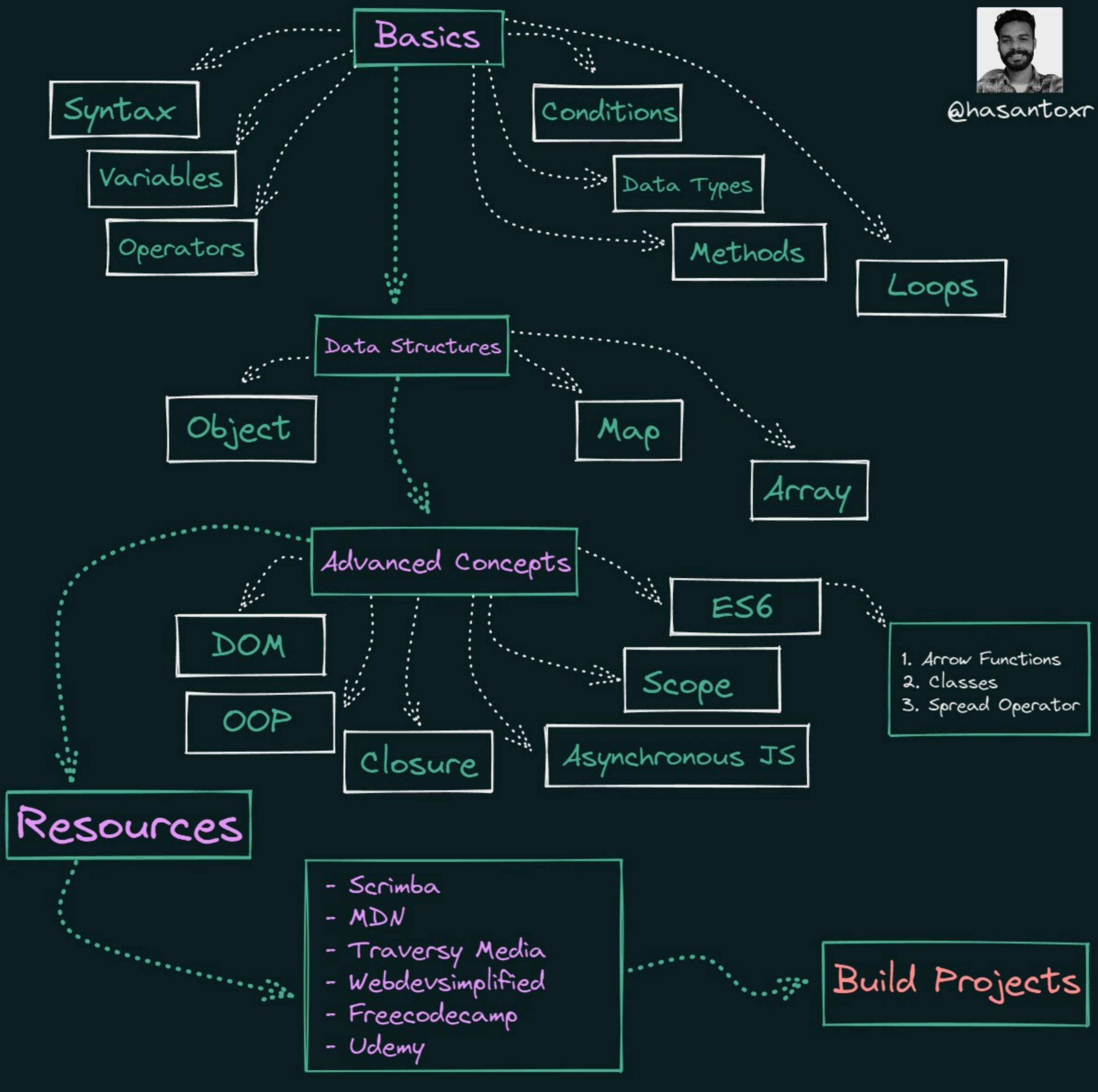


# 16. JAVASCRIPT ROADMAP CHEAT SHEET

## JavaScript Dev Roadmap



@hasantoxr



**Thank  
you!**

Do follow me on Twitter more more amazing  
Content @Hasantoxr