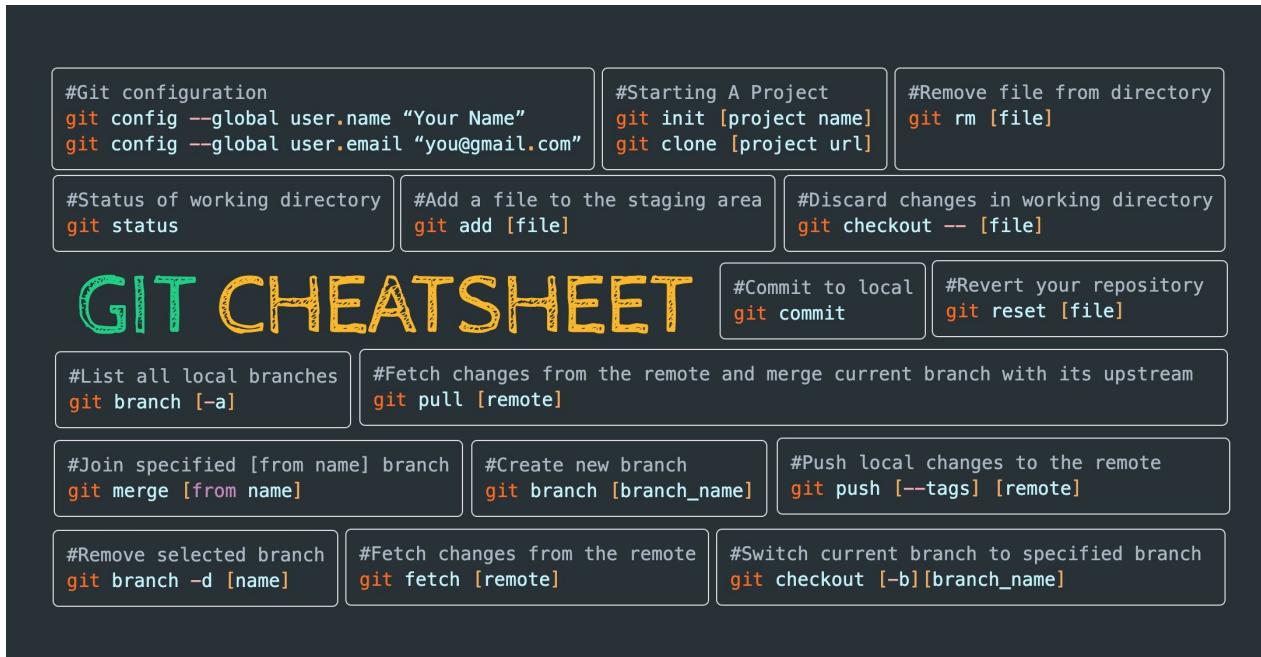


## TABLE OF CONTENTS



# 1. Git CheatSheet



A collage of various Git command snippets arranged in a grid-like pattern. The central title "GIT CHEATSHEET" is written in a large, yellow, textured font.

|   |   |  |  |
|---|---|--|--|
| #Git configuration<br><code>git config --global user.name "Your Name"</code><br><code>git config --global user.email "you@gmail.com"</code> | #Starting A Project<br><code>git init [project name]</code><br><code>git clone [project url]</code>         | #Remove file from directory<br><code>git rm [file]</code>                                  |  |
| #Status of working directory<br><code>git status</code>   | #Add a file to the staging area<br><code>git add [file]</code>  | #Discard changes in working directory<br><code>git checkout -- [file]</code>               |  |
| <b>GIT CHEATSHEET</b>   |   | #Commit to local<br><code>git commit</code>  | #Revert your repository<br><code>git reset [file]</code> |
| #List all local branches<br><code>git branch [-a]</code>  | #Fetch changes from the remote and merge current branch with its upstream<br><code>git pull [remote]</code> |  |  |
| #Join specified [from name] branch<br><code>git merge [from name]</code>  | #Create new branch<br><code>git branch [branch_name]</code>   | #Push local changes to the remote<br><code>git push [--tags] [remote]</code>               |  |
| #Remove selected branch<br><code>git branch -d [name]</code>  | #Fetch changes from the remote<br><code>git fetch [remote]</code>   | #Switch current branch to specified branch<br><code>git checkout [-b] [branch_name]</code> |  |

Credit: [Trilochan Parida](#)

# 2. Git CheatSheet with more details

## TABLE OF CONTENTS

|            |   |
|------------|---|
|            | local changes, remote commits, and untracked files.   |
| git diff   | To see the specific local changes to tracked files. Use <code>--name-only</code> to see changed filenames.                      |
| git add    | To stage changes to your tracked and untracked files. Use <code>-u</code> , <code>-a</code> , and <code>.</code> strategically. |
| git commit | To create a new commit with changes previously added. Use <code>-m</code> and add a meaningful commit message.                  |
| git push   | To send changes to your configured remote repository, most commonly GitLab or GitHub.   |

|  |   |
|--|---|
|  | and unstaged others   |
| git add -u   | add only the updated, previously-committed files  |
| git log --graph<br>--pretty=oneline<br>--abbrev-commit | for a pretty branch history. Create a shell or git alias for easy access, such as <code>git lg</code> |

| Basic Flow — Daily usage of git, including flags   |  |
|--|--|
| <code>git init</code>  | cd to your local project that you want to start versioning with git. You only have to run <code>git init</code> the first time to set up the directory for version tracking. |
| <code>git status</code><br><code>git add --all</code><br><code>git status</code><br><code>git commit -m "meaningful initial commit message"</code><br><code>git show</code>  |  |
| <code>git diff</code><br><code>git commit -a -m "Another commit message. -a performs the add step for you"</code><br><code>git status</code><br><code>git log --graph --pretty=oneline --abbrev-commit</code>  | And you begin to hack on your local files, then commit at regular intervals  |
| <code>git log --graph --pretty=oneline --abbrev-commit</code><br><code>git reset --soft HEAD~3</code><br><code>git diff --cached</code><br><code>git commit -a -m "Better commit message for last 3 commits"</code>  | After a while, you have 3 commits that would be more meaningful as a single commit   |
| <code>git status</code><br><code>git diff --cached</code><br><code>git add -u</code><br><code>git commit -m "Another commit message. -u adds updates, including deleted files"</code><br><code>git status</code><br><code>git log --graph --pretty=oneline --abbrev-commit</code><br><code>git push origin master</code> | Lastly, you delete some unneeded files in the current directory  |

**Working with a Remote Repository** — Once you get into the flow, you'll frequently contribute back to larger projects, and possibly managing forks of forks. Here are some tips for doing so.

|   |  |
|---|--|
| <code>git fetch --all</code>                                  | downloads all commits, files, and references to branches on all remote repositories so you can then <code>git checkout</code> or <code>pull</code> what you want to work on.   |
| <code>git pull --rebase &lt;remote&gt; &lt;branch&gt;</code>  | Merge all commits since your last common commit from the remote branch without creating a merge commit   |
| <code>git stash</code>  | Use this as needed to save uncommitted changes so you can <code>git stash pop</code> them onto a different branch.   |
| <code>git stash pop</code>                                    | bring it back  |
| <code>git add [-A or . or --&lt;filename&gt;]</code>          | Be intentional about what files you add to your commits, especially if you want to open a request to merge them into an upstream project.  |
| <code>git commit -m "commit message"</code>                   | Most projects have a format they prefer for commit messages. Look at <code>CONTRIBUTING.md</code> files in the project or review previous commits to get an idea of their format.  |
| <code>git push origin &lt;branch&gt;</code>                   | Push your current branch to your remote titled "origin" and branch named <code>&lt;branch&gt;</code>   |
| <code>git checkout -b &lt;new_branch&gt;</code>               | A shortcut for <code>git branch &lt;branch&gt; &amp;&amp; git checkout &lt;branch&gt;</code> . It's great for when you want to experiment with an idea and have a new branch to try it out on that can later be merged or deleted. |
| <code>git checkout master &amp;&amp; git pull --rebase</code> | Great to get to the most recent commit for a project you only infrequently follow.   |
| <code>git reset --hard origin/master</code>                   | For when you inevitably get lost in all the git-fu and need to get to a known state. <b>WARNING:</b> this erases all changes, even commits, since the last commit pushed to the remote origin on branch <code>master</code> .      |
| <code>git push origin master</code>                           | For when you inevitably do something right! Send your changes up to your remote titled <code>origin</code> on branch <code>master</code> .   |

Credit: [Trilochan Parida](#)

## 3. Javascript Operator Cheatsheet

## TABLE OF CONTENTS



|                                |                               |                         |  |
|--------------------------------|-------------------------------|-------------------------|--|
| <code>5 == 5 //true</code>     | <code>5 != 5 //false</code>   | <code>2 + 2 // 4</code> | <code>let a = 10; log(++a, a); // 11 11</code> |
| <code>5 == '5' //true</code>   | <code>5 != '5' //false</code> | <code>4 - 2 // 2</code> | <code>let b = 10; log(b++, b); // 10 11</code> |
| <code>5 === 5 //true</code>    | <code>5 !== 5 //false</code>  | <code>2 * 3 // 6</code> | <code>let c = 10; log(--c, c); // 9 9</code>   |
| <code>5 === '5' //false</code> | <code>5 !== '5' //true</code> | <code>4 / 2 // 2</code> | <code>let d = 10; log(d--, d); // 10 9</code>  |

# JAVASCRIPT OPERATOR CHEATSHEET

```
let a = 20; a += 5; // a = 25
let b = 20; b -= 5; // b = 15
let c = 20; c *= 5; // c = 100
let d = 20; d /= 5; // d = 4
let e = 20; e %= 5; // e = 0
```

```
(5 === 5) ? 'a' : 'b' // a
3 > 2 && 1 < 2 // true
3 > 2 || 5 < 2 // true
!true // false
```

```
3 > 3 // false
3 < 3 // false
3 >= 3 // true
3 <= 3 // true
5 % 2 // 1
```

Credit: [Proful Sadangi](#)

## 4. Javascript String Cheatsheet

### String declaration

```
const name = "Proful";
const city = 'Jeypore';
const msg = `${name} is
from ${city}`;
```

### String functions

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

# STRING CHEATSHEET

### String functions

```
city.replace("J", "P");
city.toUpperCase();
name.toLowerCase();
name.concat(" is good")
```

### String functions

```
name.slice(3, 6); //'ful'
city.split("y");
// [ 'Je', 'pore' ]
name.length; //6
```

Credit: [Proful Sadangi](#)

## TABLE OF CONTENTS



```
const today = new Date()
today // 2020-08-09T01:40:51.017Z
today.getDate() // 9
today.getDay() // 0 sunday
today.getMonth() // 7 (0 is jan)
today.getFullYear() // 2020
today.getYear() // 120 ?
today.getHours() // 11
today.getMinutes() // 40
today.getSeconds() // 51
today.getMilliseconds() // 24
today.getTime() // 1596937251025
today.getTimezoneOffset() // -600
```

```
today.setDate(5)
today.setMonth(5)
today.setFullYear(2022)
today.setYear(100)
today.setHours(12)
today.setMinutes(5)
today.setSeconds(8)
today.setMilliseconds(45)
today.setTime(34)
today.setUTCDate(11)
today.setUTCMonth(5)
today.setUTCFullYear(2022)
today.setUTCHours(5)
today.setUTCMinutes(10)
today.setUTCSeconds(24)
today.setUTCMilliseconds(22)
```

```
Date.parse('9 Aug 2020') // 1596895200000
// Returns number
// milliseconds since January 1, 1970
// Returns NaN if invalid
```

```
today.getUTCDate() // 9
today.getUTCDay() // 0
today.getUTCMonth() // 7
today.getUTCFullYear() // 2020
today.getUTCHours() // 1
today.getUTCMinutes() // 40
today.getUTCSeconds() // 51
today.getUTCMilliseconds() // 31
```

```
const today = new Date()
today.toString() // 'Sun Aug 09 2020 16:44:29 GMT+1000'
// (Australian Eastern Standard Time)
today.toLocaleString() // '09/08/2020, 16:44:29'
today.toGMTString() // 'Sun, 09 Aug 2020 06:44:29 GMT'
today.toUTCString() // 'Sun, 09 Aug 2020 06:44:29 GMT'
```

Credit: [Proful Sadangi](#)

## 6. Javascript Object Cheatsheet

**Object Declaration**

```
const twit = {
  name: "Proful",
  follower: 4817,
  1: "hi"
}
// converted to string
// storing keyvalue pairs.
// data unordered
// keys are unique
```

**Dot Notation**

```
twit.name // "Proful"
twit.follower // 4817
twit.follower.count
// Accessing nested props
```

**Square Notation**

```
twit['name'] // "Proful"
// Can be dynamic/variable
```

**Empty Object creations**

```
const person = {}
const person = new Object()
```

**Pass by reference**

```
delete twit.name // both key & value
twit.randomKey // undefined
twit.follower = 5000
// declared as const but mutable
for(const key in twit) {
  console.log(key) // name
  // followers
}
```

**getter**

```
const twit = {
  name: "Proful",
  get profile() {
    return `Hi ${this.name.toLowerCase()}`
  },
  set profile(prof) {
    this.name = "Mr " + prof
  }
}
twit.profile // 'Hi proful'
twit.profile = 'Steve'
twit.name // 'Hi Steve'
```

**this refer to twit object**

```
const twit = {
  name: "Proful",
  hi() {
    console.log(`Hi ${this.name}`)
  },
  hello: () => {
    console.log(`Hello ${twit.name}`)
  }
}
// You cannot use this here
```

## TABLE OF CONTENTS



# 7. Javascript Array Cheatsheet

```
['a', 'b'].concat(['c']) //['a','b','c']
['a', 'b'].join(~) //a~b
['a','b','c'].slice(1) //['b', 'c' ]
['a','b','b'].indexOf('b') // 1
['a','b','b'].lastIndexOf('b') //2
```

```
['a','b','c'].forEach(x => console.log(x))
[1,2,3].every(x => x < 10 )//true
[1,2,3].some(x => x < 2 )//true
[1,2,3].filter(x => x < 2 )//[1]
```

## ARRAY CHEATSHEET

```
[1, 2, 3].map(x => x * 2 )//[2, 4, 6]
[1, 2, 3].reduce((x,y) => x * y)//6
[2, 15,3].sort()// [ 15, 2, 3 ] 😊
[1, 2, 3].reverse()//[ 3, 2, 1 ]
[1, 2, 3].length//3
```

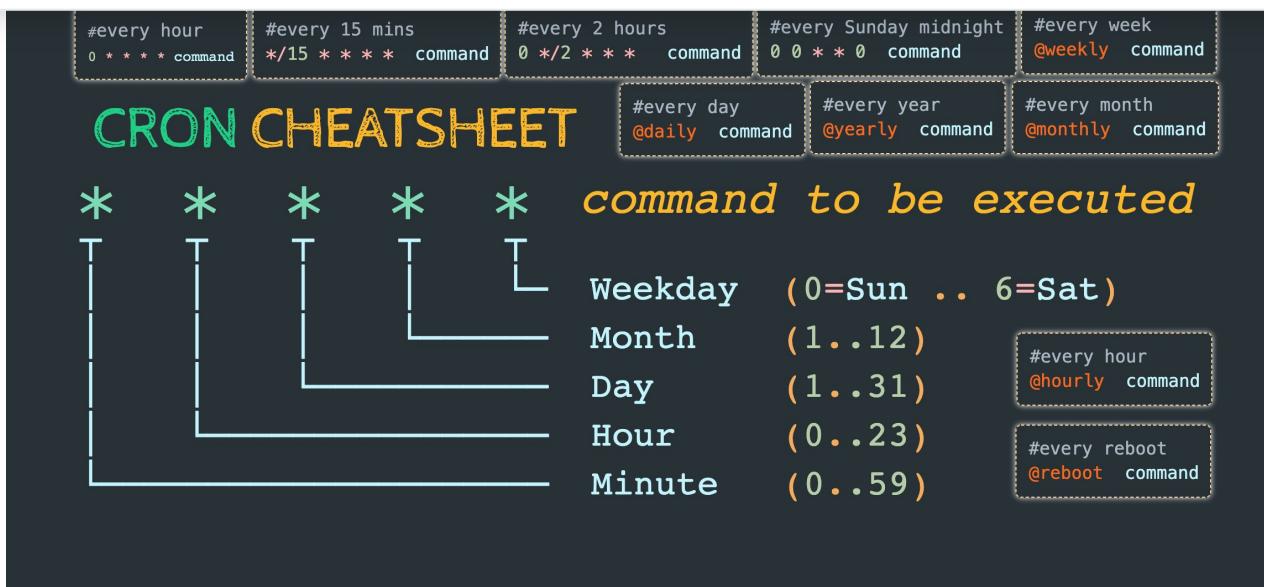
```
const arr = [1, 2, 3]
const x=arr.shift()//arr=[ 2, 3 ],x=1
const x=arr.unshift(9)//arr=[ 9,1,2,3 ],x=4
const x=arr.pop()//arr=[ 1, 2 ],x=3
const x=arr.push(5)//arr=[1,2,3,5],x=4
```

```
const arr=['a','b','c','d'];const mod = arr.splice(1,2,'z');//arr=['a','z','d'],mod=['b','c']
```

Credit: [Proful Sadangi](#)

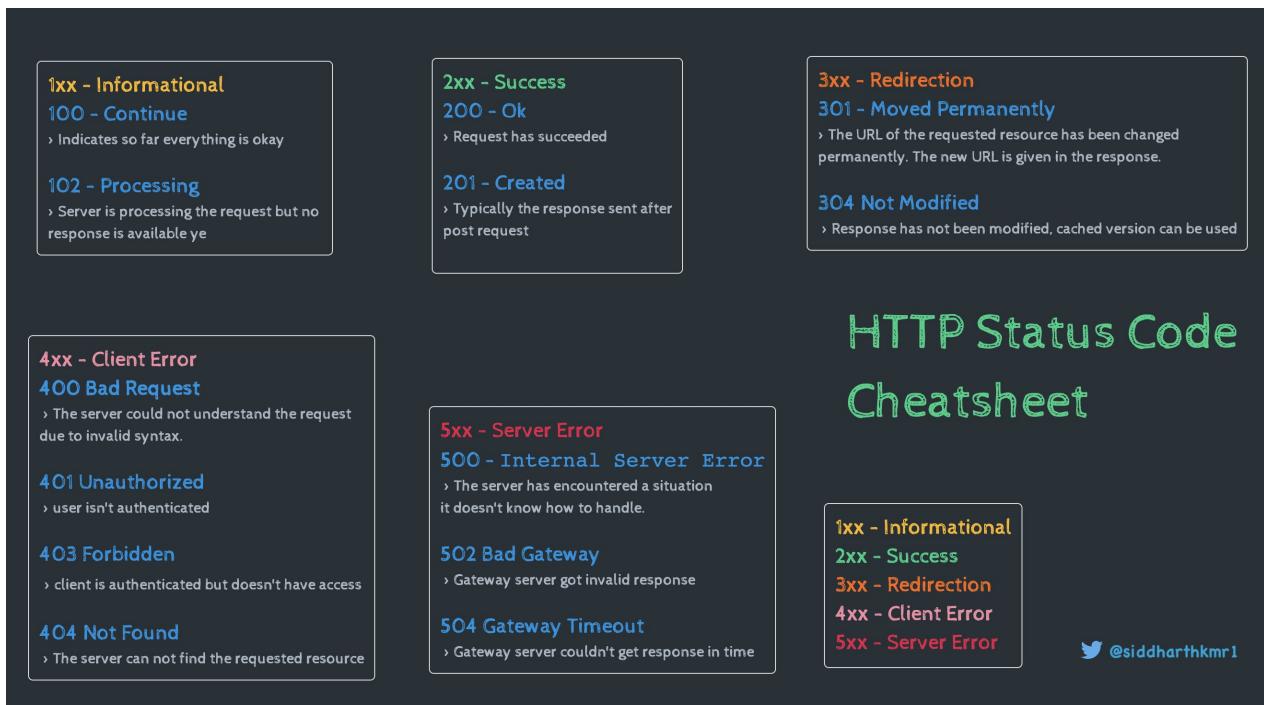
# 8. Cron Cheatsheet

## TABLE OF CONTENTS



Credit: [Trilochan Parida](#)

## 9. HTTP Status Code Cheatsheet



## TABLE OF CONTENTS



# 10. Javascript DOM Cheatsheet

```
//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('I ❤ it')

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

//insert h2 before h1
app.insertBefore(h2, h1)
```

(p>I ❤ it/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
```

**JAVASCRIPT DOM**

 profulsadangi

Credit: [Proful Sadangi](#)

# 11. CSS Selectors Cheatsheet

## TABLE OF CONTENTS

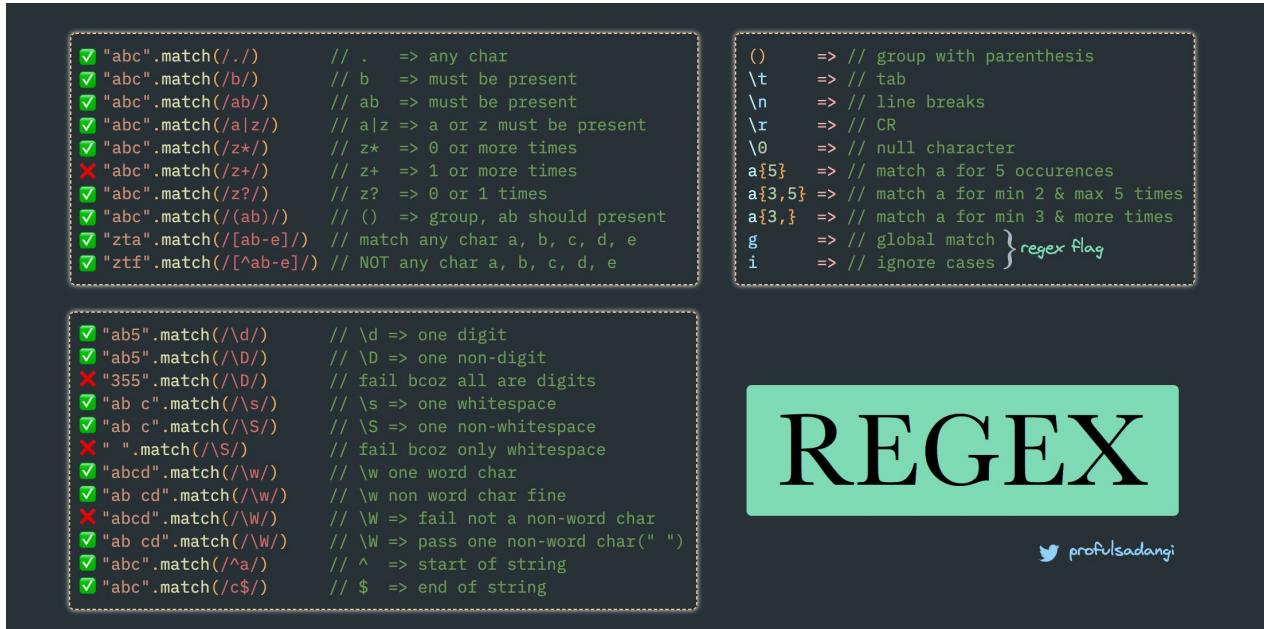


**CSS SELECTORS**

|  |   |  |                                   |                                   |
|--|---|--|-----------------------------------|-----------------------------------|
| #Id Selector<br>#name -Element with id="firstname" | #Element Selector<br>p -Selects all <p> elements  | #Selects All Unvisited Links<br>a:link                   |                                   |                                   |
| #Descendant<br>.parent .child                      | #Direct Descendant<br>.parent > .child            | #Adjacent Sibling<br>.child + .sibling                   | #Far Sibling<br>.child ~ .sibling | #Disabled Input<br>input:disabled |
| #Selects On Mouse Over<br>a:hover                  |   | #Active Link<br>a:active                                 | #Checked Input<br>input:checked   |                                   |
| #Selects All Visited Links<br>a:visited            | #Insert Content Before Every Element<br>p::before | #Element Without Children<br>p:empty                     |                                   |                                   |
| #Elements With a Target Attribute<br>[target]      | #Elements With target="_blank"<br>[target=_blank] | #Elements Has Word "flower"<br>[title~=flower]           |                                   |                                   |
| #Element Begins With "https"<br>[href^="https"]    | #Element Ends With ".pdf"<br>[href\$=".pdf"]      | #Element contains the substring "tech"<br>[href*="tech"] |                                   |                                   |

Credit: [Trilochan Parida](#)

## 12. Javascript Regex Cheatsheet



**REGEX**

|   |   |
|---|---|
| <ul style="list-style-type: none"> <li>✓ "abc".match(/./) // . =&gt; any char</li> <li>✓ "abc".match(/b/) // b =&gt; must be present</li> <li>✓ "abc".match(/ab/) // ab =&gt; must be present</li> <li>✓ "abc".match(/a z/) // a z =&gt; a or z must be present</li> <li>✓ "abc".match(/z*/) // z* =&gt; 0 or more times</li> <li>✗ "abc".match(/z+/) // z+ =&gt; 1 or more times</li> <li>✓ "abc".match(/z?/) // z? =&gt; 0 or 1 times</li> <li>✓ "abc".match(/(ab)/) // () =&gt; group, ab should present</li> <li>✓ "zta".match(/[^ab-e]/) // match any char a, b, c, d, e</li> <li>✓ "ztf".match(/[^ab-e]/) // NOT any char a, b, c, d, e</li> </ul>  | <ul style="list-style-type: none"> <li>( ) =&gt; // group with parenthesis</li> <li>\t =&gt; // tab</li> <li>\n =&gt; // line breaks</li> <li>\r =&gt; // CR</li> <li>\0 =&gt; // null character</li> <li>a{5} =&gt; // match a for 5 occurrences</li> <li>a{3,5} =&gt; // match a for min 2 &amp; max 5 times</li> <li>a{3,} =&gt; // match a for min 3 &amp; more times</li> <li>g =&gt; // global match } regex flag</li> <li>i =&gt; // ignore cases</li> </ul> |
| <ul style="list-style-type: none"> <li>✓ "ab5".match(/\d/) // \d =&gt; one digit</li> <li>✓ "ab5".match(/\D/) // \D =&gt; one non-digit</li> <li>✗ "355".match(/\D/) // fail bcoz all are digits</li> <li>✓ "ab c".match(/\s/) // \s =&gt; one whitespace</li> <li>✓ "ab c".match(/\S/) // \S =&gt; one non-whitespace</li> <li>✗ " ".match(/\S/) // fail bcoz only whitespace</li> <li>✓ "abcd".match(/\w/) // \w one word char</li> <li>✓ "ab cd".match(/\w/) // \w non word char fine</li> <li>✗ "abcd".match(/\W/) // \W =&gt; fail not a non-word char</li> <li>✓ "ab cd".match(/\W/) // \W =&gt; pass one non-word char(" ")</li> <li>✓ "abc".match(/^a/) // ^ =&gt; start of string</li> <li>✓ "abc".match(/c\$/) // \$ =&gt; end of string</li> </ul> |   |

Credit: [Proful Sadangi](#)

## TABLE OF CONTENTS



# Command Line Cheat Sheet

|  |   |  |  |
|--|---|--|--|
| <code>pwd</code><br>Check the Current Directory  | <code>ls</code><br>Display List of Files                                    | <code>rm file_to_remove</code><br>Remove Files & Directories                                     | <code>touch [file_name]</code><br>Create Files |
| <code>cd ..</code><br>Parent Directory   | <code>mkdir</code><br>Create a Directory                                    | <code>cd</code><br>Move Between Directories  | <code>clear</code><br>Clear Screen             |
| <code>mv old_file_name new_file_name</code><br><code>mv old_directory_name new_directory_name</code><br>Rename Files & Directories | <code>ls -a</code><br>list all files including hidden file starting with '' |  @iamjatinrao |  |

Credit: [Jatin Rao](#)

## 14. VSCode Keyboard Shortcuts CheatSheet



# TABLE OF CONTENTS

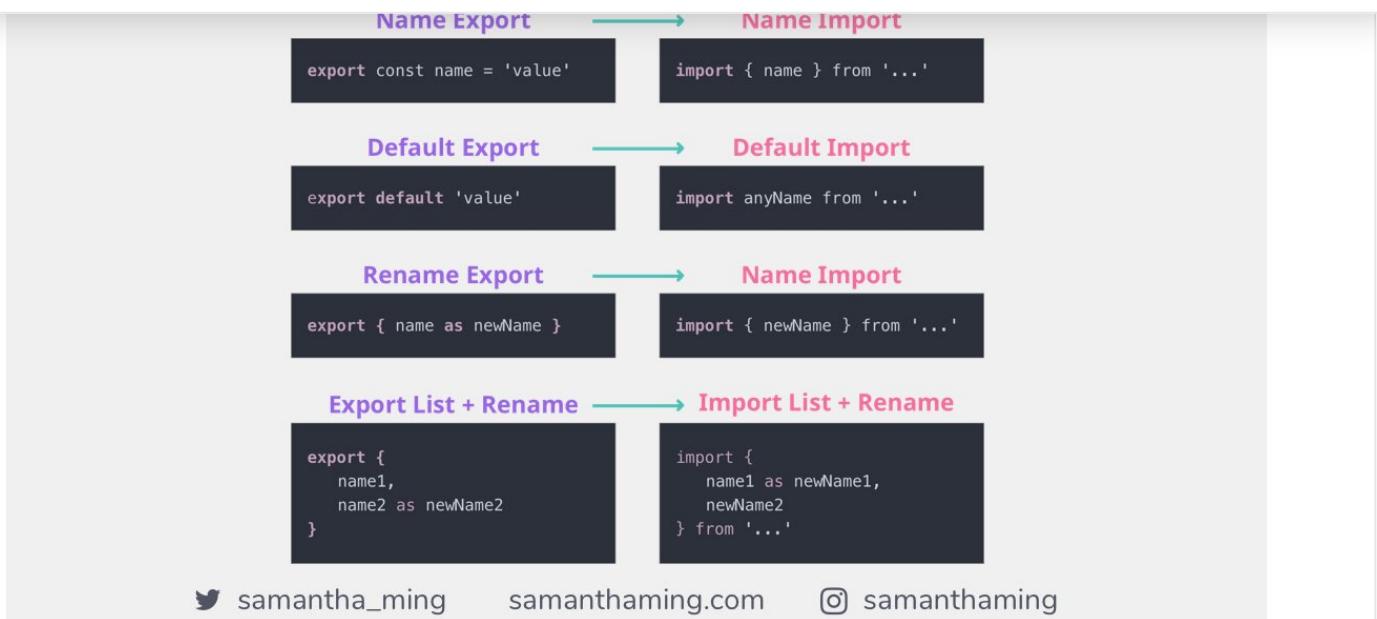


| Keyboard shortcuts for Windows   |  |
|--|--|
| <b>General</b>   |  |
| <b>Ctrl+Shift+P, F1</b>  | Show Command Palette   |
| <b>Ctrl+P</b>  | Quick Open, Go to File...  |
| <b>Ctrl+Shift+N</b>  | New window-instance  |
| <b>Ctrl+Shift+W</b>  | Close window-instance  |
| <b>Ctrl+,</b>  | User Settings  |
| <b>Ctrl+K Ctrl+S</b>   | Keyboard Shortcuts   |
| <b>Basic editing</b>   |  |
| <b>Ctrl+X</b>  | Cut line (empty selection)   |
| <b>Ctrl+C</b>  | Copy line (empty selection)  |
| <b>Alt+↑ / ↓</b>   | Move line up/down  |
| <b>Shift+Alt+↑ / ↓</b>   | Copy line up/down  |
| <b>Ctrl+Shift+K</b>  | Delete line  |
| <b>Ctrl+Enter</b>  | Insert line above  |
| <b>Ctrl+Shift+Enter</b>  | Insert line below  |
| <b>Ctrl+Shift+\`</b>   | Jump to matching bracket   |
| <b>Ctrl+[ / ]</b>  | Indent/outdent line  |
| <b>Home / End</b>  | Go to beginning/end of line  |
| <b>Ctrl+Home</b>   | Go to beginning of file  |
| <b>Ctrl+End</b>  | Go to end of file  |
| <b>Ctrl+↑ / ↓</b>  | Scroll line up/down  |
| <b>Alt+PgUp / PgDn</b>   | Scroll page up/down  |
| <b>Ctrl+Shift+[</b>  | Fold (collapse) region   |
| <b>Ctrl+Shift+]</b>  | Unfold (uncollapse) region   |
| <b>Ctrl+K Ctrl+[</b>   | Fold (collapse) all subregions   |
| <b>Ctrl+K Ctrl+]</b>   | Unfold (uncollapse) all subregions   |
| <b>Ctrl+K Ctrl+0</b>   | Fold (collapse) all regions  |
| <b>Ctrl+K Ctrl+J</b>   | Unfold (uncollapse) all regions  |
| <b>Ctrl+K Ctrl+C</b>   | Add line comment   |
| <b>Ctrl+K Ctrl+U</b>   | Remove line comment  |
| <b>Ctrl+/</b>  | Toggle line comment  |
| <b>Shift+Alt+A</b>   | Toggle block comment   |
| <b>Alt+Z</b>   | Toggle word wrap   |
| <b>Navigation</b>  |  |
| <b>Ctrl+T</b>  | Show all Symbols   |
| <b>Ctrl+G</b>  | Go to Line...  |
| <b>Ctrl+P</b>  | Go to File...  |
| <b>Ctrl+Shift+O</b>  | Go to Symbol...  |
| <b>Ctrl+Shift+M</b>  | Show Problems panel  |
| <b>F8</b>  | Go to next error or warning  |
| <b>Shift+F8</b>  | Go to previous error or warning  |
| <b>Ctrl+Shift+Tab</b>  | Navigate editor group history  |
| <b>Alt+← / →</b>   | Go back / forward  |
| <b>Ctrl+F</b>  | Find   |
| <b>Ctrl+H</b>  | Replace  |
| <b>F3 / Shift+F3</b>   | Find next/previous   |
| <b>Alt+Enter</b>   | Select all occurrences of Find match   |
| <b>Ctrl+D</b>  | Add selection to next Find match   |
| <b>Ctrl+K Ctrl+D</b>   | Move last selection to next Find match                                       |
| <b>Alt+C / R / W</b>   | Toggle case-sensitive / regex / whole word                                   |
| <b>Multi-cursor and selection</b>  |  |
| <b>Alt+Click</b>   | Insert cursor  |
| <b>Ctrl+Alt+↑ / ↓</b>  | Insert cursor above / below  |
| <b>Ctrl+U</b>  | Undo last cursor operation   |
| <b>Shift+Alt+I</b>   | Insert cursor at end of each line selected                                   |
| <b>Ctrl+L</b>  | Select current line  |
| <b>Ctrl+Shift+L</b>  | Select all occurrences of current selection                                  |
| <b>Ctrl+Shift+T</b>  | Select all occurrences of current word                                       |
| <b>Shift+Alt+→</b>   | Expand selection   |
| <b>Shift+Alt+←</b>   | Shrink selection   |
| <b>Shift+Alt+ (drag mouse)</b>   | Column (box) selection   |
| <b>Ctrl+Shift+Alt+ (arrow key)</b>   | Column (box) selection   |
| <b>Shift+Alt+Alt+ (arrow key)</b>  | Column (box) selection page up/down  |
| <b>+PgUp/PgDn</b>  |  |
| <b>Rich languages editing</b>  |  |
| <b>Ctrl+Space</b>  | Trigger suggestion   |
| <b>Ctrl+Shift+Space</b>  | Trigger parameter hints  |
| <b>Shift+Alt+F</b>   | Format document  |
| <b>Ctrl+K Ctrl+F</b>   | Format selection   |
| <b>F12</b>   | Go to Definition   |
| <b>Alt+F12</b>   | Peek Definition  |
| <b>Ctrl+K F12</b>  | Open Definition to the side  |
| <b>Ctrl+.</b>  | Quick Fix  |
| <b>Shift+F12</b>   | Show References  |
| <b>F2</b>  | Rename Symbol  |
| <b>Ctrl+K Ctrl+X</b>   | Trim trailing whitespace   |
| <b>Ctrl+K M</b>  | Change file language   |
| <b>Editor management</b>   |  |
| <b>Ctrl+F4, Ctrl+W</b>   | Close editor   |
| <b>Ctrl+K F</b>  | Close folder   |
| <b>Ctrl+\`</b>   | Split editor   |
| <b>Ctrl+1 / 2 / 3</b>  | Focus into 1 <sup>st</sup> , 2 <sup>nd</sup> or 3 <sup>rd</sup> editor group |
| <b>Ctrl+K Ctrl+--/--</b>   | Focus into previous/next editor group  |
| <b>Ctrl+Shift+PgUp / PgDn</b>  | Move editor left/right   |
| <b>Ctrl+K --/--</b>  | Move active editor group   |
| <b>Ctrl+S</b>  | Save   |
| <b>Ctrl+Shift+S</b>  | Save As...   |
| <b>Ctrl+K S</b>  | Save All   |
| <b>Ctrl+F4</b>   | Close  |
| <b>Ctrl+K Ctrl+W</b>   | Close All  |
| <b>Ctrl+Shift+T</b>  | Reopen closed editor   |
| <b>Ctrl+K Enter</b>  | Keep preview mode editor open  |
| <b>Ctrl+Tab</b>  | Open next  |
| <b>Ctrl+Shift+Tab</b>  | Open previous  |
| <b>Ctrl+K P</b>  | Copy path of active file   |
| <b>Ctrl+K R</b>  | Reveal active file in Explorer   |
| <b>Ctrl+K O</b>  | Show active file in new window-instance                                      |
| <b>Display</b>   |  |
| <b>F11</b>   | Toggle full screen   |
| <b>Shift+Alt+0</b>   | Toggle editor layout (horizontal/vertical)                                   |
| <b>Ctrl+= / -</b>  | Zoom in/out  |
| <b>Ctrl+B</b>  | Toggle Sidebar visibility  |
| <b>Ctrl+Shift+E</b>  | Show Explorer / Toggle focus   |
| <b>Ctrl+Shift+F</b>  | Show Search  |
| <b>Ctrl+Shift+G</b>  | Show Source Control  |
| <b>Ctrl+Shift+D</b>  | Show Debug   |
| <b>Ctrl+Shift+X</b>  | Show Extensions  |
| <b>Ctrl+Shift+H</b>  | Replace in files   |
| <b>Ctrl+Shift+J</b>  | Toggle Search details  |
| <b>Ctrl+Shift+U</b>  | Show Output panel  |
| <b>Ctrl+Shift+V</b>  | Open Markdown preview  |
| <b>Ctrl+K V</b>  | Open Markdown preview to the side  |
| <b>Ctrl+K Z</b>  | Zen Mode (Esc Esc to exit)   |
| <b>Debug</b>   |  |
| <b>F9</b>  | Toggle breakpoint  |
| <b>F5</b>  | Start/Continue   |
| <b>Shift+F5</b>  | Stop   |
| <b>F11 / Shift+F11</b>   | Step into/out  |
| <b>F10</b>   | Step over  |
| <b>Ctrl+K Ctrl+I</b>   | Show hover   |
| <b>Integrated terminal</b>   |  |
| <b>Ctrl+`</b>  | Show integrated terminal   |
| <b>Ctrl+Shift+`</b>  | Create new terminal  |
| <b>Ctrl+C</b>  | Copy selection   |
| <b>Ctrl+V</b>  | Paste into active terminal   |
| <b>Ctrl+↑ / ↓</b>  | Scroll up/down   |
| <b>Shift+PgUp / PgDn</b>   | Scroll page up/down  |
| <b>Ctrl+Home / End</b>   | Scroll to top/bottom   |
| Other operating systems' keyboard shortcuts and additional unassigned shortcuts available at <a href="https://aka.ms/vscodekeybindings">aka.ms/vscodekeybindings</a> |  |

Credit: [Visual Studio Code Tips and Tricks](#)

## 15. Module Cheatsheet

## TABLE OF CONTENTS



Credit: [Samantha Ming](#)

## 16. Array Methods Cheatsheet

## TABLE OF CONTENTS

@sulco

|  |                                    |   |
|--|------------------------------------|---|
|  | <code>.map( □ → ○ )</code>         | → |
|  | <code>.filter( □ )</code>          | → |
|  | <code>.find( □ )</code>            | → |
|  | <code>.findIndex( □ )</code>       | → |
|  | <code>.fill( 1, ○ )</code>         | → |
|  | <code>.copyWithin( 2, 0 )</code>   | → |
|  | <code>.some( □ )</code>            | → |
|  | <code>.every( □ )</code>           | → |
|  | <code>.reduce( acc + curr )</code> | → |

Credit: [Tomek Sułkowski](#)

## Wrap Up