

GIT commands

<code>git init</code>	initialize an existing directory
<code>git clone [url]</code>	retrieve an entire repository
<code>git status</code>	show modified files in working directory
<code>git add [file]</code>	add a file as it looks now to your next commit
<code>git reset [file]</code>	unstage a file while retaining the changes in working directory
<code>git diff</code>	diff of what is changed but not staged
<code>git commit -m " [commit message]"</code>	commit your staged content as a new commit snapshot
<code>git branch</code>	list your branches, current has a *
<code>git branch [branch name]</code>	create a new branch at the current commit
<code>git checkout</code>	switch to another branch
<code>git merge [branch name]</code>	merge the specified branch into current one
<code>git log</code>	show all commits
<code>git push</code>	transmit local branch commits to remote repo
<code>git pull</code>	fetch and merge any commits from the remote repo

Regex

<code>^</code> or <code>\A</code>	Start of string
<code>\$</code> or <code>\Z</code>	end of string
<code>\b</code>	word boundary
<code>\B</code>	Not word boundary
<code>\s</code>	white space
<code>\S</code>	not white space
<code>?=</code>	look ahead
<code>?!</code>	negative lookahead
<code>*</code>	0 or more
<code>+</code>	1 or more
<code>?</code>	0 or 1
<code>{3}</code>	exactly 3
<code>{3,}</code>	3 or more
<code>{3,5}</code>	3, 4, or 5
<code>[a-z]</code>	lowercase a - z
<code>[A-Z]</code>	uppercase A-Z
<code>[^abc]</code>	not (a or b or c)
<code>[0-7]</code>	digits from 0 to 7
<code>(a C)</code>	a or C

JavaScript String Objects

	length	property
	<code>charAt(n)</code>	method
	<code>charCodeAt(n)</code>	method
<code>let a = " 10"; let b = 20; console.log(a + b)</code>		1020
<code>let a = " 10"; let b = 20; console.log(+a + b)</code>		30
<code>let a = " 10"; let b = 20; console.log(a - b)</code>		-10
<code>let a = " 10"; let b = 20; console.log(a * b)</code>		200
<code>let a = "CST82 85"; let b = 20; console.log(a - b)</code>		NaN

PHP - database connection

MYSQli connection
<code>\$dbc = mysqli_connect(- \$host, \$user, \$pw, \$db);</code>
Query
<code>\$result = mysqli_query(- (\$dbc, \$query);</code>
Close
<code>mysqli_close(\$dbc);</code>

JavaScript Math Objects

<code>Math.round(x)</code>	Math.r - x to nearest integer
<code>Math.pow(x, y)</code>	Math.p - x to the power of y
<code>Math.sqrt(x)</code>	Math.s - square root of x
<code>Math.abs(x)</code>	Math.a - absolute x
<code>Math.random()</code>	Math.r - random number between 0 (inclusive) to 1(exclusive)

JavaScript Arrays

<code>array.sort(function(a, b) {return a-b})</code>	array.s or t (function(a, b) {return a-b})
<code>array.sort((a,b) =>{ a-b})</code>	array.s or t ((a,b) =>{ a-b})
<code>a-b = positive</code>	b sorted before value a
<code>a-b = negative</code>	a sorted before value b
<code>a-b = 0</code>	items stay in the same spot
<code>array.filter(function(element) {return (comparison)})</code>	array.f il ter (function(element) {return (comparison)})
<code>array.filter((element) => {return (comparison)});</code>	array.f il ter ((element) => {return (comparison)});
<code>array.map((item) => - arr ay.k ey);</code>	array.m ap - new array containing only items with key
<code>array.push(value)</code>	array.p us h - add value to end of array



JavaScript Arrays (cont)

`array.pop()` remove value at end of array, returning value

`array.unshift(value)` add value at start of array

`array.shift()` remove value at start of array, returning value

`array.splice(start, deleteElement, ...valuesToAdd)` add or remove elements anywhere in the array and returns deleted elements if any

PHP functions

`isset()` test if variable exists

`empty()` tests for empty variable

`mysqli_fetch_array($result)` fetch each row of query \$result

`exit()` end script immediately

`trim($string)` remove leading and trailing whitespace

`file($fn)` read contents of \$fn, returns array of strings

`file_get_contents($fn)` returns single string

`file_put_contents($fn, data, mode)` write content from data to \$fn, using mode

`file_exists($fn)` checks if \$fn exists

`scandir($path)` returns array of lists of files and directories if \$path is valid, else false

PHP functions (cont)

`glob()` returns array of all file names that match pattern, with path

`getcwd()` returns string of current working dir

Cookies & Session

Cookies

- limited in size by browser
- stored client-side
- can be disabled

Sessions

- limited by server space
- stored server-side
- expire when browser closed
- cannot be modified by users

PHP Session Commands

`session_start()` start a session

`$_SESSION["variableName"]` store session variable

`unset($_SESSION["variableName"])` remove session variable

`session_destroy()` destroy session and all session variables

JSON

easy to parse and generate

can contain both object and array structures

contain key and value pairs

keys are always strings

values can be string, number, Boolean, null, object, or array

HAS no version, very stable, strictly in UNICODE

Example object: `$json = { "Name": "Alex", "Age": 37, "Admin": true } ;`

JSON in PHP

`json_encode()` convert PHP array to JSON array

`JSON_FORCE_OBJECT` - changes above to result in indexed array

`json_decode($json, true)` JSON string to PHP array

`json_decode($json, false)` (default) JSON string to PHP object

`JSON.stringify()` take JS object and produce JSON string

`JSON.parse()` take JSON string and produce JS object

PHP Error Reporting

`error_reporting` types of errors reported

`display_errors` whether messages displayed in browser

`log_errors` on server or somewhere else

var | let | const

VAR	LET	CONST
function-scoped	block-scoped	block-scoped
updatable	updatable	not updatable
redecl-able	not redecl-able	not redecl-able
hoisted null	not hoisted	



By WildRyc

cheatography.com/wildryc/

Not published yet.

Last updated 10th December, 2022.

Page 2 of 2.

Sponsored by [Readable.com](https://readable.com)

Measure your website readability!

<https://readable.com>