OneStop(Nerd)Shop Cheat Sheet

by pastrami (pastrami) via cheatography.com/140937/cs/31147/

Javascript

let <name> = <value>	Declare variable
let <name> = prompt("<text">)	Prompt user for input
+	Concat or add
if (<condition>){<if true>} else {<if false>}	Conditional
<condition> ? <if true> : <if false>	Conditional shorthand
<variable>.trim()	Trim whitespace
let <objname> = {<name> : <value>, <name> : <value>}	Create object with attributes
||	Logical or
<variable>.replace("<toreplace>","<replacewith>")	Replace in string
&&	Logical and
<variable>.toUpperCase()	To uppercase
let <name> = [<var1>,<var2>]	Create list / Array
listName[<index>]	Access value in index position of array
"<string> $(variable)"	Literals, add value to string. Can concat too with +
listName[<index>] = <value>	Update value in index position of array
Math.round(<value>)	Round number
Math.floor(<number>)	Round down
Math.ceil(<number>)	Round up
Math.min(<num1>,<num2>)	Lowest value provided
Math.max(<num1>,<num2>)	Highest value provided
let <name> = Date()	Current time
<name>.getMinutes()	Return value. Works for hours, date, day, month, year as well
Date.parse("")	Create date
while (condition) {<stuff to do>}	Loop while condition is true
<name>.forEach(function(<name>) {<stuff to do}	Do once for each item in list

Javascript (cont)

do {<stuff to do>} while(-<condition>)	Do while condition is true
for (<initial>,<condition->,<increament>) {stuff to do}	For loop while condition is true. Increamenet runs after each loop.
function <name>(<things to bring in>) {<thing to do>}	Create a callable function
function <name>(<variable> = <default>) { }	Provide a default value to variable if one isn't provided
function <name>() {<thing to do> return <thing to give back>}	Returns value
let <name> = new <objName>()	Create object instance
<objName>.<variable>	Call variable value inside object. New variables can be used to create new prop
let <name> = [{<objVar>:<objVal>},{<objVar>:<objVal>}]	Array of objects
for(let <var> in <obj>) {<thing to do>}	Enumerate through object properties
let <name> = document.querySelector(#"text")	Applies to the below, returns element / name on page matching selector
<name>.addEventListener("<listenerType>",-<function>())	Execute function when `<te xt>` is selected. Type can be click, mouseenter, mouseleave, mousedown, mouseup, mousemove, keydown, keyup
let <name> = document.querySelectorAll(#"-text")	Returns all elements / name on page matching selector
<form> <input type ="<type>" id="<name>" /> </form> <script> <function to do> </script>	Create a form with function
export <item>	Export for use elsewhere

Python

**	Exponent
%	Modulus (Remainder)
//	Integer Division
/	Division
*	Multiplication, can replace strings
-	Subtraction
+	Addition, can concatenate strings
<name> = <value>	Declare variable
_<name> = <value>	"Unuseful" variable
#	Comment
"""<text>"""	Docstring / multi line comment
print(<value>)	Prints value to console
<name> = input()	Assign input from user to variable
len(<value>)	Determines length
str(<value>)	Converts to string
int(<value>)	Converts to int
float(<value>)	Convert to float
==	Equal to
!=	Not equal to
<, >, <=, >=	Less than, greater than
<value> is <boolean>	Implicint boolean evaluation
(<condition>) and/or (<condition>)	Mix boolean and comparison
if <condition>: <thing to do> elif <condition>: <thing to do> else: <thing to do>	Runs if, if true. Tries elif if not. Runs else if none are true.
while <condition>: <thing to do>	Does while the condition is true
while<condition>: <thing to do> break	Stops when break is reached
while <condition>: <thing to do> continue	Jumps to start of loop when continue is reached
for <thing> in <list>: <do this>	Do for each item in list
for <thing> in range(<number>)	Do number of times in range. Range can take a (<start>,<stop>,<interation increase>). Negative counts down.

Python (cont)

for <thing> in <list>: <thing to do> break else: <thing to do>	When break is reached, else will run
import <module name>, <modulename>	Imports modules
from <module name> import <item>	Imports specific section of module
sys.exit()	Ends program
def <name>(<stuff>,<to>,<bring>) : <stuff to do>	

SQL

Coming soon

Shell

cd	Navigation, ~ for home, .. for up
mkdir	Creates directory
rm	Removes file (-rf for all/dir)
ls	List contents, -R for sub dir's, -l for permissions
pwd	Show current path
cat	Create file
mv	Renames dir
sudo	Superuser/root command
history	Command history
pr	File edit, -x for columns, -h header, -n line numbers
Chown <user>	r(ead), w(rite), (e)x(ecute), -= for none. <user>:<group> filename for dir
adduser	Create user
passwd -l	Change password
usermod -a -G <group> <name>	Add user to group
deluser <name> <group>	Remove from group
userdel	Remove user
finger	Shows logged in users
ssh -p <port> <user>@<ip>	SSH into ip at port
fg	Run stopped process in foreground

OneStop(Nerd)Shop Cheat Sheet

by pastrami (pastrami) via cheatography.com/140937/cs/31147/

Shell (cont)

bg	Send process to background
top	Shows active processes
ps	Shows process running for user
kill PID	Kill process
df	Shows hard disk space
free	Shows RAM
nano	Editor
curl	Download from URL
tar -C <path>	Unzip to path
find <path> -name <name>	Find file in path, can use wildcard *
systemctl status	Check service status
systemctl stop	Stop service
systemctl start	Start service
systemctl restart	Restart service
service --status-all	Show all service statuses
scp <source> <destination>:<-path>	Move files through ssh
mv <source> <destination>	Move file

Screen

ctrl-a	Use screen shortcut
-S <name>	Create and name sessions
c	Create window
"	List windows
0-9	Switch to window #
A	Rename window
S	split horizontally
|	Split vertically
<tab>	Switch focus
?	List commands
-list	List screens
-r	Resume screen

Screen (cont)

ctrl-a	Toggle screens
Q	close all but current
X	Close current

Multipass

launch --name <name>	Creates and starts new instance
exec <instance> -- <command>	Sends command
list	List instances
Stop	
Start	
Delete	
shell <instance>	Enter instance

Extras

openvpn <file_locati-on>	start ovpn session
SecList	Common used everything - found here
admin'#	Example of injection, ends with ' then starts a comment with #. Sometimes any pw can work after
Wappalyzer	Site tech scanner

admin:admin
guest:guest
user:user
root:root
administrator:password

nmap

-sV	Probe ports for service/version
-sC	Scans with default set of scripts (**INTRUSIVE**)

gobuster

Requires Go	go get && go build && go install
dir	Specify web directory enumeration
--url	Target
--wordlist	Wordlist to use
-x	Search for specific file extensions

https://github.com/OJ/gobuster.git

ftp

anonymous	Username sometimes works
get	Downloads file
bye	Exits

impacket

python3 mssqlclient.py <destination>/<user>@<ip> -windows-auth	
mssqlclient.py	
SELECT is_srvrolemember('sys-admin');	Checks curernt privledge
EXEC xp_cmdshell 'net user';	Check if command shell is active

Found here.
Python classes for working with network protocols

wifite

--dict	Specify dictionary for use
--kill	Kill conflicting processes
WPS Pixie-Dust attack	
WPS PIN attack	
PMKID capture	
WPA Handshake capture	

mySQL

-u	Specify user
-h	Connect to host

smbclient

-L	List directories
\\\\\<ip>\\<folder>	smb to folder
Empty pw	Can work for guest access
get	Downloads content of dir
exit	Closes
-N	No password

telnet

Sometimes passwords can be blank	
Common usernames	root, administrator, admin, root