# Sommario

[PHP 1](#_Toc36500010)

[JavaScript 1](#_Toc36500011)

[Java 1](#_Toc36500012)

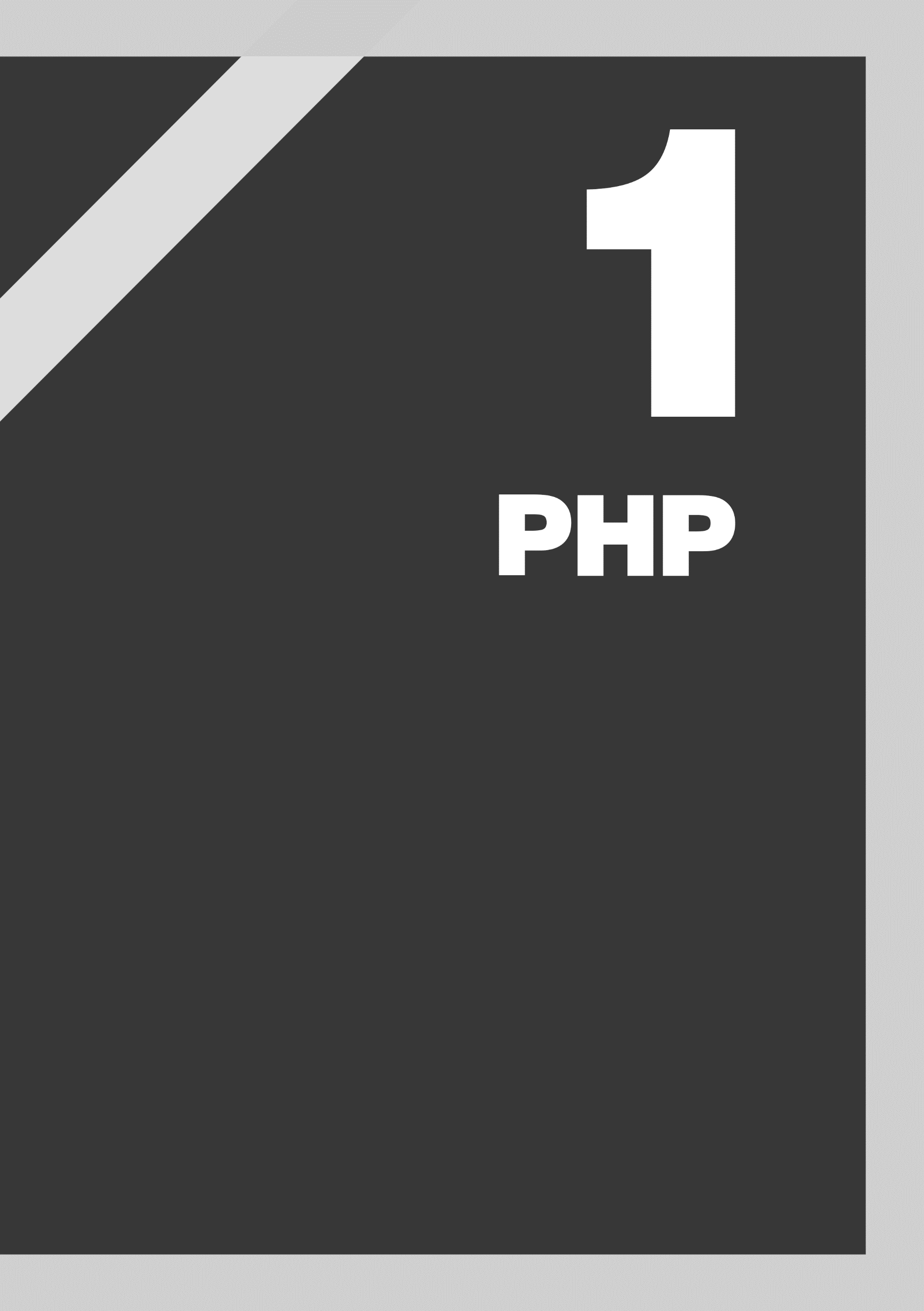
[MySQL 1](#_Toc36500013)

[HTML 1](#_Toc36500014)

[CSS 1](#_Toc36500015)

[GIT 1](#_Toc36500016)

[Sistemi e reti 1](#_Toc36500017)



Sommario

[**Commenti** 2](#_Toc36497792)

[**Importazione di un file:** 2](#_Toc36497793)

[**Modificatori di accesso:** 2](#_Toc36497794)

[**Dichiarazione variabile** 2](#_Toc36497795)

[**Dichiararazione costante** 2](#_Toc36497796)

[**Casting** 3](#_Toc36497797)

[**Eseguire una stampa** 3](#_Toc36497798)

[**Concatenazione stringhe** 3](#_Toc36497799)

[**Defizione di un metodo** 3](#_Toc36497800)

[**Dichiarazione e Istanza Array** 4](#_Toc36497801)

[**Dichiarazione e Istanza Array Associativo** 4](#_Toc36497802)

[**Creazione di una Classe** 4](#_Toc36497803)

[**Definizione costruttore** 5](#_Toc36497804)

[**Istanza della classe** 5](#_Toc36497805)

[**If, Else If, Else** 5](#_Toc36497806)

[**Ciclo While, Do-While, For e ForEach** 6](#_Toc36497807)

[**Creazione di un oggetto mysqli** 6](#_Toc36497808)

[**Eseguire una query** 7](#_Toc36497809)

[**Ottenere un valore in GET** 7](#_Toc36497810)

[**Ottenere un valore in POST** 7](#_Toc36497811)

[**Criptare una password** 8](#_Toc36497812)

[**Verificare correttezza di una password** 8](#_Toc36497813)

[**Ottenere informazioni di una variabile** 9](#_Toc36497814)

[**Ottenere il valore intero da una variabile** 9](#_Toc36497815)

[**Determina se una variabile è dichiara e diversa da NULL** 10](#_Toc36497816)

[**Determina se una variabile è vuota** 10](#_Toc36497817)

[**Genera l’hash di un valore** 11](#_Toc36497818)

**PHP:**

# **Commenti**

// Commento su una linea

/\*

Commento multi linea

\*/

# **Importazione di un file:**

include “nomefile.php”

require\_once “nomefile.php”

require “nomefile.php”

include\_once “nomefile.php”

# **Modificatori di accesso:**

public

private

protected

# **Dichiarazione variabile**

**Sintassi**

$nomeVariabile = valore

# **Dichiararazione costante**

**Sintassi**

const COSTANTE = valore

define(“COSTANTE”, “valore”)

# **Casting**

**Sintassi**

(tipo) $variabile;

# **Eseguire una stampa**

**Sintassi**

**Parametri**

echo($arg1, ...);

$arg1: mixed

... : string (opzionale)

# **Concatenazione stringhe**

**Sintassi**

“stringa1”.” ”.”stringa2”.” ”.”stringa3”

# **Defizione di un metodo**

**Sintassi**

function nomeFunzione(*parametri, ...*){

// Codice

}

# **Dichiarazione e Istanza Array**

**Sintassi**

**Parametri**

$array **= array**($*val1, ...*)

**Valore restituito**

$val1: mixed (opzionale)

... : mixed (opzionale)

Array

# **Dichiarazione e Istanza Array Associativo**

**Sintassi**

**Parametri**

$array **= array**(*key1 => $val1, ...)*

**Valore restituito**

$key1: string (opzionale)

$val1: mixed (opzionale)

... : mixed (opzionale)

Array

# **Creazione di una Classe**

**Sintassi**

class NomeClasse{

/\* Codice \*/

}

# **Definizione costruttore**

**Sintassi**

function \_\_construct(...) {

/\* Codice \*/

}

# **Istanza della classe**

**Sintassi**

$oggetto = new NomeClasse(...)

# **If, Else If, Else**

**Sintassi**

if (condizione){

/\* Codice \*/

}elseif (condizione){

/\* Codice \*/

}else{

/\* Codice \*/

}

# **Ciclo While, Do-While, For e ForEach**

**Sintassi**

while(condizione){

/\* Codice \*/

}

do{

/\* Codice \*/

}while(condizione)

for (inizializzazione; condizione; incrementazione|decrementazione){

/\* Codice \*/

}

foreach($Array as $ValoreSingolo){

/\* Codice \*/

}

# **Creazione di un oggetto mysqli**

**Sintassi**

**Parametri**

new **mysqli**($host, $username, $password, $dbname, *$port, $socket*);

**Valore restituito**

$host: string

$username: string

$password: string | NULL

$dbname: string

*$port*: integer *(opzionale)*

*$socket*: string *(opzionale)*

Object | False

# **Eseguire una query**

**Sintassi**

**Parametri**

$mysqli->query($query, *$resultmode = MYSQLI\_STORE\_RESULT*)

**Valore restituito**

$query: string

*$resultmode*: int (opzionale)

mixed | False

# **Ottenere un valore in GET**

**Sintassi**

$\_GET[‘nome’]

# **Ottenere un valore in POST**

**Sintassi**

$\_POST[‘nome’]

# **Criptare una password**

**Sintassi**

**Parametri**

password\_hash($string, $algo, *$options*)

**Valore restituito**

$string: string

*$algo*: int (opzionale)

*$options*: array (opzionale)

string

# **Verificare correttezza di una password**

**Sintassi**

**Parametri**

password\_verify($password, $hash)

**Valore restituito**

$password: string

$hash: string

bool

# **Ottenere informazioni di una variabile**

**Sintassi**

**Parametri**

var\_dump($var, *...*)

$var: string

$hash: string

# **Ottenere il valore intero da una variabile**

**Sintassi**

**Parametri**

**intval**($var, $*basePerLaConversione = 10*);

**Valore restituito**

$var: mixed

$basePerLaConversione : int (opzionale)

int

# **Determina se una variabile è dichiara e diversa da NULL**

**Sintassi**

**Parametri**

**isset**($var, *...*)

**Valore restituito**

$var: mixed

... : mixed (opzionale)

bool

# **Determina se una variabile è vuota**

**Sintassi**

**Parametri**

**empty**($var)

**Valore restituito**

$var: mixed

bool

# **Genera l’hash di un valore**

**Sintassi**

**Parametri**

hash($algo, $string, *$raw\_output = FALSE*)

**Valore restituito**

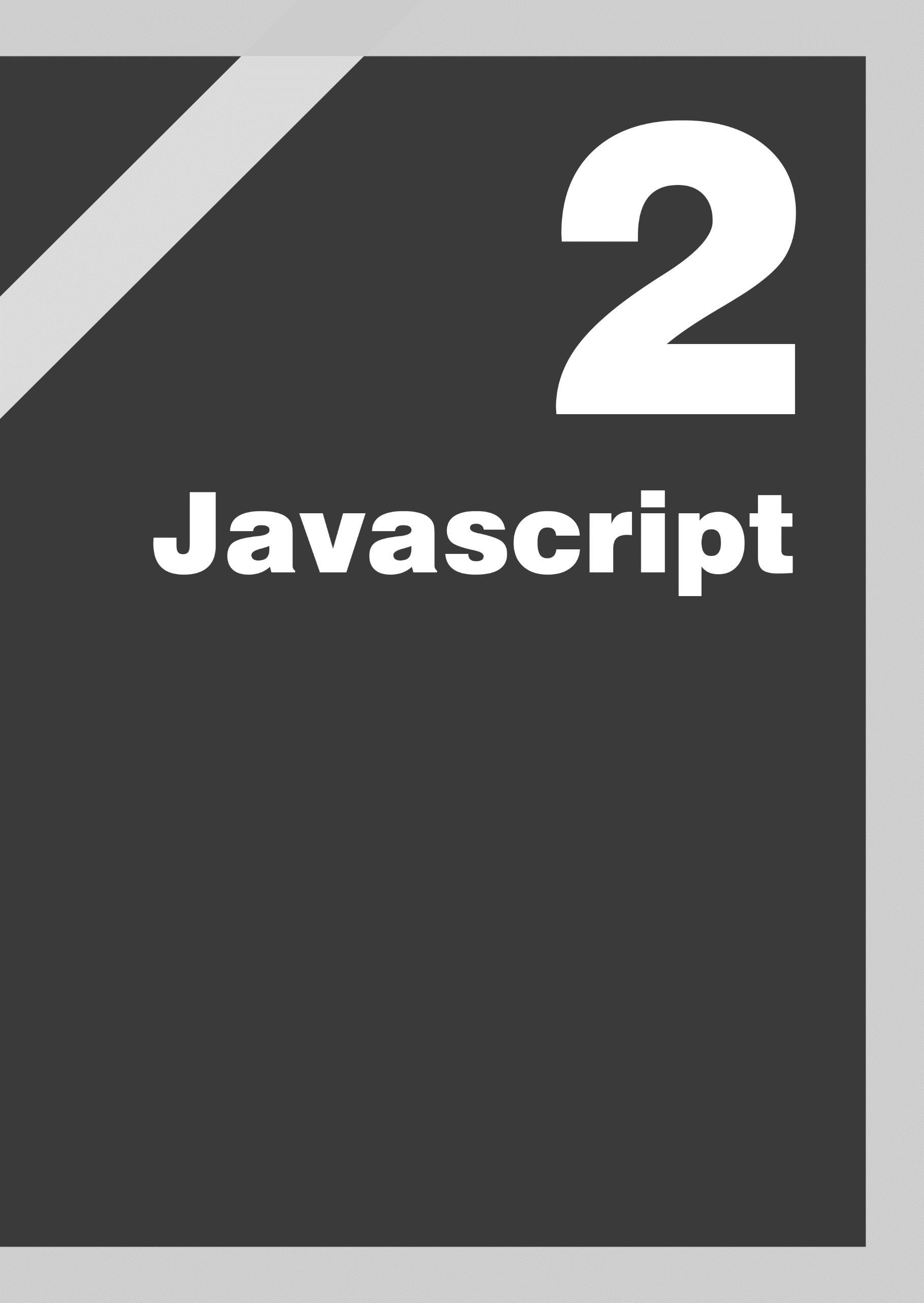
$algo: string

$string: string

*$raw\_output*: bool (opzionale)

string

|  |
| --- |
|  |
|  |  |



Sommario

[**Commenti** 18](#_Toc36498790)

[**Dichiarazione variabili** 18](#_Toc36498791)

[**Dichiarazione costanti** 18](#_Toc36498792)

[**Casting** 18](#_Toc36498793)

[**Eseguire una stampa (su Console)** 18](#_Toc36498794)

[**Concatenazione Stringhe** 19](#_Toc36498795)

[**Creazione di un metodo** 19](#_Toc36498796)

[**Dichiarazione Array** 19](#_Toc36498797)

[**Dichiarazione Oggetto** 19](#_Toc36498798)

[**Dichiarazione Classe** 20](#_Toc36498799)

[**Dichiarazione Costruttore** 20](#_Toc36498800)

[**Istanza della Classe** 20](#_Toc36498801)

[**If, Else If e Else** 20](#_Toc36498802)

[**Cicli While, Do-While, For e ForEach** 21](#_Toc36498803)

[**Aggiunta di un elemento a un array:** 21](#_Toc36498804)

[**Aggiunta di un elemento in un Array in una posizione determinata** 22](#_Toc36498805)

[**Ottenere la posizione di un determinato elemento in un Array** 22](#_Toc36498806)

[**Creazione di un oggetto Date** 23](#_Toc36498807)

[**Ottenere il giorno della settimana da un oggetto Date** 23](#_Toc36498808)

[**Ottenere il numero del giorno da un oggetto Date** 24](#_Toc36498809)

[**Ottenere il numero del mese da un oggetto Date** 24](#_Toc36498810)

[**Ottenere l’anno da un oggetto Date** 24](#_Toc36498811)

[**Ottenere uno specifico elemento del DOM** 25](#_Toc36498812)

[**Ottenere più elementi specifici dal DOM** 25](#_Toc36498813)

[**Creare un elemento DOM** 26](#_Toc36498814)

[**Aggiunta di un elemento nel DOM** 26](#_Toc36498815)

[**Aggiunta di un evento a un elemento del DOM** 26](#_Toc36498816)

**JavaScript:**

# **Commenti**

// Commento su una linea

/\*

Commento multi linea

\*/

# **Dichiarazione variabili**

**Sintassi**

let nomeVariabile;

# **Dichiarazione costanti**

**Sintassi**

const nomeVariabile = valore;

# **Casting**

**Sintassi**

Tipo(valore)

# **Eseguire una stampa (su Console)**

**Sintassi**

**Parametri**

console.log(obj1, ...)

obj1: mixed

*...*: mixed

# **Concatenazione Stringhe**

**Sintassi**

“Stringa1” + “ ” + “Stringa2”

# **Creazione di un metodo**

const | let nomeFunzione = (*nomeParametro*, *...*) => {

/\* Codice \*/

}

**Sintassi**

function nomeFunzione(*nomeParametro,* *...*){

/\* Codice \*/

}

# **Dichiarazione Array**

**Sintassi**

let array = []

# **Dichiarazione Oggetto**

**Sintassi**

let object = { /\* Codice \*/ }

# **Dichiarazione Classe**

**Sintassi**

class NomeClasse{

/\* Codice \*/

}

# **Dichiarazione Costruttore**

**Sintassi**

function NomeClasse(*...*){

/\* Codice \*/

}

# **Istanza della Classe**

**Sintassi**

let | const object = new NomeClasse(*...*);

# **If, Else If e Else**

**Sintassi**

if(condizione){

/\* Codice \*/

}else if (condizione){

/\* Codice \*/

}else(condizione){

/\* Codice \*/

}

# **Cicli While, Do-While, For e ForEach**

**Sintassi**

while(condizione){

/\* Codice \*/

}

do{

/\* Codice \*/

}while(condizione)

for(inizializzazione; condizione; incrementazione|decrementazione){

/\* Codice \*/

}

array.forEach(singoloElemento => {

/\* Codice \*/

});

# **Aggiunta di un elemento a un array:**

**Sintassi**

**Parametri**

array.**push**(arg1, ..., argN)

**Valore restituito**

argN: mixed

Number

# **Aggiunta di un elemento in un Array in una posizione determinata**

**Sintassi**

array[posizione] = valore

# **Ottenere la posizione di un determinato elemento in un Array**

**Sintassi**

**Parametri**

array.**indexOf**(elementoDaCercare, ***aPartireDall’Indice***)

**Valore restituito**

elementoDaCercare: mixed

*aPartireDall’Indice*: Number (opzionale)

Number

# **Creazione di un oggetto Date**

**Sintassi**

**Parametri**

new **Date**();

new **Date**(value);

new **Date**(dateString);

new **Date**(year, monthIndex, *day*, *hours, minutes, seconds, milliseconds*);

**Valore restituito**

value: number

dateString: string

year: number

monthIndex: number

*day*: number (opzionale)

*hours*: number (opzionale)

*minutes*: number (opzionale)

*seconds* number (opzionale)

*milliseconds* number (opzionale)

Object

# 

# **Ottenere il giorno della settimana da un oggetto Date**

**Sintassi**

oggettoDate.**getDay**()

**Valore restituito**

Number

# **Ottenere il numero del giorno da un oggetto Date**

**Sintassi**

oggettoDate.**getDate**()

**Valore restituito**

Number

# **Ottenere il numero del mese da un oggetto Date**

**Sintassi**

oggettoDate.**getMonth**()

**Valore restituito**

Number

# 

# **Ottenere l’anno da un oggetto Date**

**Sintassi**

oggettoDate.**getFullYear**()

**Valore restituito**

Number

# **Ottenere uno specifico elemento del DOM**

**Sintassi**

**Parametri**

document.**querySelector**(elementoDaTrovare)

**Valore restituito**

elementoDaTrovare: string

Object

# **Ottenere più elementi specifici dal DOM**

**Sintassi**

**Parametri**

document.**querySelectorAll**(elementoDaTrovare)

**Valore restituito**

elementoDaTrovare: string

NodeList

# **Creare un elemento DOM**

**Sintassi**

**Parametri**

document.**createElement**(nomeTagHTML, options)

**Valore restituito**

nomeTagHTML: string

options: object (opzionale)

Object

# **Aggiunta di un elemento nel DOM**

**Sintassi**

**Parametri**

document.body.**append**(oggettoNodo)

oggettoElement: object

# **Aggiunta di un evento a un elemento del DOM**

**Sintassi**

**Parametri**

elementoDOM.**addEventListener**(evento, funzione);

evento: string

funzione: function

****

**Sommario**

[**Commento** 28](#_Toc36284774)

[**Importazione di un file** 28](#_Toc36284775)

[**Modificatore di accesso** 28](#_Toc36284776)

[**Dichiarazione variabile** 28](#_Toc36284777)

[**Dichiarazione costante** 28](#_Toc36284778)

[**Casting di una variabile** 28](#_Toc36284779)

[**Prendere un dato in input( importare java.util.Scanner)** 29](#_Toc36284780)

[**Stampa** 29](#_Toc36284781)

[**Concatenazione stringhe** 29](#_Toc36284782)

[**Definizione metodo** 29](#_Toc36284783)

[**Dichiarazione Array** 29](#_Toc36284784)

[**Dichiarazione ArrayList** 30](#_Toc36284785)

[**Dichiarazione classe** 30](#_Toc36284786)

[**Dichiarazione classe Main** 30](#_Toc36284787)

[**Dichiarazione costruttore** 30](#_Toc36284788)

[**Dichiarazione istanza della classe** 30](#_Toc36284789)

[**If, else, else if** 31](#_Toc36284790)

[**while, do while, for, for each** 31](#_Toc36284791)

[**Istanziare un oggetto Scanner** 32](#_Toc36284792)

[**Istanziare un oggetto Date** 33](#_Toc36284793)

**Java**

# **Commento**

/\*

Commento multilinea

\*/

// Commento su una riga

# **Importazione di un file**

import file;

# **Modificatore di accesso**

**Sintassi**

public

private

protected

# **Dichiarazione variabile**

**Sintassi**

type variable;

# **Dichiarazione costante**

**Sintassi**

final type COSTANT;

# **Casting di una variabile**

**Sintassi**

(type) variable;

# **Prendere un dato in input( importare java.util.Scanner)**

**Sintassi**

Scanner scan = new Scanner(System.in); //Istanza dello Scanner

int variable = scan.nextInt(); //Prende in input una variabile int

float variable = scan.nextFloat(); //Prende in input una variabile float

double variable = scan.nextDouble(); //Prende in input una variabile double

String variable = scan.next(); //Prende in input una variabile String

# **Stampa**

**Sintassi**

System.out.println(variable);

# **Concatenazione stringhe**

**Sintassi**

variable = “Mario”+”Elisa”

# **Definizione metodo**

**Sintassi**

public|private|protected type|void nameMethod(*type var1, type var2, …*){

/\* codice \*/

}

# **Dichiarazione Array**

**Sintassi**

type[] nameArray;

# **Dichiarazione ArrayList**

**Sintassi**

import java.util.ArrayList

ArrayList<type> nameArrayList = new ArrayList<type>();

# **Dichiarazione classe**

**Sintassi**

public|private|protected class nameClass{

/\* codice \*/

}

}

# **Dichiarazione classe Main**

**Sintassi**

public static void main(String[] args) {

/\* codice \*/

}

}

# **Dichiarazione costruttore**

**Sintassi**

public Nameclass(*parameters*){

/\* codice \*/

}

# **Dichiarazione istanza della classe**

**Sintassi**

Nameclass variable= new Nameclass();

# **If, else, else if**

**Sintassi**

if (condition){

/\* codice \*/

} else {

/\* codice \*/

} else if (condition){

/\* codice \*/

}

# **while, do while, for, for each**

**Sintassi**

while (condition){

/\* codice \*/

}

do {

/\* codice \*/

} while(condition);

for(initialization; condition; increment|decrement){

/\* codice \*/

}

for(type var: array){

/\* codice \*/

}

# **Istanziare un oggetto Scanner**

*Istanziare un oggetto Scanner permette di ottenere i dati che l’utente deve inserire usando la propria tastiera come Input*

**Sintassi**

**Parametri**

new **Scanner**(source)

new **Scanner**(source, charsetName);

**Valore restituito**

source: File | InputStream | Path | Readable | ReadableByteChannel | String

source: File | InputStream | Path | Readable | ReadableByteChannel | String

charsetName: String

Scanner

# **Istanziare un oggetto Date**

*Istanziare un oggetto Date permette di eseguire operazioni sulle date senza dover effettuare manualmente le conversioni*

Object

milliseconds: long

**Valore restituito**

**Parametri**

new **Date**();

new **Date**(milliseconds);

**Sintassi**



**Sommario**

[**DML (Data Manipulation Language)** 37](#_Toc36286111)

[**Insert :** 37](#_Toc36286112)

[**Update :** 37](#_Toc36286113)

[**Delete :** 37](#_Toc36286114)

[**QL (Query Language)** 38](#_Toc36286115)

[**Select :** 38](#_Toc36286116)

[**Operatori BETWEEN e LIKE :** 38](#_Toc36286117)

[**Operatori IN e IS NULL :** 38](#_Toc36286118)

[**Join :** 39](#_Toc36286119)

[**Funzioni di aggregazione :** 40](#_Toc36286120)

[**COUNT :** 40](#_Toc36286121)

[**AVG :** 40](#_Toc36286122)

[**MAX :** 40](#_Toc36286123)

[**MIN :** 40](#_Toc36286124)

[**SUM :** 40](#_Toc36286125)

[**ORDER BY :** 41](#_Toc36286126)

[**GROUP BY e HAVING :** 41](#_Toc36286127)

[**Subquery :** 41](#_Toc36286128)

[**DCL (Data Control Language)** 42](#_Toc36286129)

[**COMMIT :** 42](#_Toc36286130)

[**ROLLBACK :** 42](#_Toc36286131)

[**SAVEPOINT :** 42](#_Toc36286132)

[**GRANT :** 42](#_Toc36286133)

[**REVOKE :** 42](#_Toc36286134)

[**RECOVER TABLE :** 42](#_Toc36286135)

[**CHECK TABLE :** 42](#_Toc36286136)

[**REPAIR TABLE :** 42](#_Toc36286137)

[**DROP VIEW :** 42](#_Toc36286138)

[**CREATE VIEW :** 42](#_Toc36286139)

[**DDL (Data Definition Language)** 43](#_Toc36286140)

[**CREATE :** 43](#_Toc36286141)

[**ALTER:** 43](#_Toc36286142)

[**DROP :** 43](#_Toc36286143)

[**TRUNCATE :** 43](#_Toc36286144)

[**RENAME :** 43](#_Toc36286145)

[**Tabelle:** 44](#_Toc36286146)

[**CREATE TABLE :** 44](#_Toc36286147)

[**ALTER TABLE :** 44](#_Toc36286148)

[**DROP TABLE :** 44](#_Toc36286149)

[**TRUNCATE TABLE :** 44](#_Toc36286150)

[**RENAME TABLE :** 44](#_Toc36286151)

# **DML (Data Manipulation Language)**

## **Insert :**

*L’Istruzione Insert viene utilizzata per inserire nuovi record in una tabella.*

INSERT INTO table\_name (column1, column2, column3, …)

VALUES (value1, value2, value3, …);

## **Update :**

*L’Istruzione Update viene utilizzata per modificare i record esistenti in una tabella.*

UPDATE table\_name

SET column1 = value1, column2 = value2, …

WHERE condition;

## **Delete :**

*L’Istruzione Delete viene utilizzata per eliminare i record esistenti in una tabella.*

INSERT INTO nomeTabella (colonna1, colonna2, colonna3, …)

VALUES (value1, value2, value3, …);

# **QL (Query Language)**

## **Select :**

*Il costrutto Select è composto da:.*

* *\_Target list*
* *\_FROM clause*
* *\_WHERE clause*

*Sia nella target list che nella from possono essere inseriti gli alias.*

SELECT column1

FROM table1 AS t1

WHERE condizione;

## **Operatori BETWEEN e LIKE :**

*L’operatore BETWEEN serve per verificare che un attributo sia in un certo intervallo.*

SELECT column1

FROM table1 AS t1

WHERE t1.valore BETWEEN 160 && 170;

*L’operatore LIKE serve per effettuare una ricerca su stringhe o caratteri simili.*

SELECT column1,column2

FROM table1 AS t1

WHERE t1.column1 LIKE R\* || t1.column2 LIKE G\*;

## **Operatori IN e IS NULL :**

*L’operatore IN confronta un attributo con i valori contenuti all’interno dell’operando.*

SELECT column1

FROM table1 AS t1

WHERE t1.column1 IN (1, 2, 3);

*L’operatore IS NULL serve per vedere se ho un’istanza con un attributo nullo.*

SELECT column1

FROM table1 AS t1

WHERE t1.column1 IS NULL;

## **Join :**

*Il costrutto Join serve a combinare i dati di due o più tabelle basandosi su alcune condizioni logiche.*

*Esistono diversi tipi di Join:*

* *INNER JOIN prende solo i record in comune.*
* *LEFT JOIN prende anche i record non comuni della prima tabella nominata*
* *RIGHT JOIN prende anche i record non comuni della seconda tabella nominata*
* *FULL JOIN prende tutti i record non comuni alle tabelle*

SELECT column1, column2

FROM table1 AS t1 INNER JOIN table2 AS t2 ON t1.column1 = t2.column1

WHERE condition;

SELECT column1, column2

FROM table1 AS t1, table2 as t2

WHERE t1.column1 = t2.column1 && condition;

# **Funzioni di aggregazione :**

## **COUNT :**

*L’operatore COUNT serve per contare le istanze che ha un’entità (conta anche i valori null).*

SELECT COUNT(column1) AS tot

FROM table\_name;

## **AVG :**

*L’operatore AVG serve per calcolare la media.*

SELECT AVG(column1) AS average

FROM table\_name;

## **MAX :**

*L’operatore MAX restituisce il valore più alto contenuto nella colonna in questione.*

SELECT MAX(column1) AS max

FROM table\_name;

## **MIN :**

*L’operatore MIN restituisce il valore più basso contenuto nella colonna in questione.*

SELECT MIN(column1) AS min

FROM table\_name;

## **SUM :**

*L’operatore SUM restituisce la somma dei valori contenuti in una colonna.*

SELECT MIN(column1) AS min

FROM table\_name;

## **ORDER BY :**

*La parola chiave ORDER BY viene utilizzata per ordinare il set di risultati in ordine crescente o decrescente. Per ordinare i record in ordine decrescente, utilizzare la parola chiave DESC.*

SELECT column1, column2, …

FROM table\_name

ORDER BY column1, column2, … ASC|DESC;

## **GROUP BY e HAVING :**

*L’istruzione GROUP BY raggruppa righe con gli stessi valori in righe di riepilogo, l’istruzione GROUP BY viene spesso utilizzata con funzioni aggregate per raggruppare il set di risultati per una o più colonne.*

SELECT column\_name(s)

FROM table\_name

WHERE condition

GROUP BY column\_name(s);

*La clausola HAVING è stata aggiunta a SQL perché non è stato possibile utilizzare la parola chiave WHERE con le funzioni di aggregazione. Si usa solo quando è presente una GROUP BY.*

SELECT column\_name(s)

FROM table\_name

WHERE condition

GROUP BY column\_name(s)

HAVING condition

ORDER BY column\_name(s);

## **Subquery :**

*Le Subquery o query annidate sono semplicemente delle query dentro altre query.*

SELECT column\_name(s)

FROM table\_name

WHERE condition > (SELECT column\_name(s)

FROM table\_name

WHERE condition);

# **DCL (Data Control Language)**

## **COMMIT :**

*con la COMMIT le modifiche effettuate in una transazione diventano permanenti.*

## **ROLLBACK :**

*Operazione opposta a quella della COMMIT, riporta lo stato del database all’ultima COMMIT eseguita, annullando, tutte le operazioni effettuate dopo quel momento.*

## **SAVEPOINT :**

*Specificare un punto all’interno di una transazione alla quale poi sarà possibile effettuare un ROLLBACK.*

## **GRANT :**

*per concedere il permesso( es. Inserimento in tabella, modifica o eliminazione ).*

## **REVOKE :**

*per revocare il permesso( es. cancellazione di una tabella ).*

## **RECOVER TABLE :**

*consente di recuperare una tabella da una copia di sicurezza.*

## **CHECK TABLE :**

*controlla la corrispondenza dei dati tra le tabelle e i suoi indici.*

## **REPAIR TABLE :**

*Ricostruisce gli indici quando il comando CHECK non ha funzionato.*

## **DROP VIEW :**

*elimina lista.*

## **CREATE VIEW :**

*crea lista.*

# **DDL (Data Definition Language)**

## **CREATE :**

*Per creare database e oggetti simili (tabella, indice, viste, procedura di archiviazione, funzione e trigger)***.**

## **ALTER:**

*Modifica la struttura del database esistente.*

## **DROP :**

*Cancella gli oggetti dal database.*

## **TRUNCATE :**

*Rimuove tutti i record da una tabella, inclusi tutti gli spazi allocati per i record.*

## **RENAME :**

*Rinomina un oggetto.*

# **Tabelle:**

## **CREATE TABLE :**

CREATE TABLE table\_name(

Attr1 tipodiDato(grandezza),

Attr2 tipodiDato(grandezza),

… ,

primary key(attrChiave),

foreign key(attrMigrato) references table\_name2(attrMigrato));

## **ALTER TABLE :**

ALTER TABLE table\_name ADD COLUMN attr3 tipodiDato(grandezza);

## **DROP TABLE :**

DROP TABLE table\_name

## **TRUNCATE TABLE :**

TRUNCATE TABLE table\_name

## **RENAME TABLE :**

RENAME TABLE old\_name TO new\_name



**Sommario**

[**Commenti** 47](#_Toc36284603)

[**Intestazione HTML** 47](#_Toc36284604)

[**Attributi Tag:** 47](#_Toc36284605)

[**Sezione HTML** 47](#_Toc36284606)

[**Sezione Head** 47](#_Toc36284607)

[**Importazione di un file (CSS, Font, etc …)** 47](#_Toc36284608)

[**Sezione Body** 47](#_Toc36284609)

[**Titolo a grandezza variabile N, da 1 a 6** 48](#_Toc36284610)

[**Paragrafo** 48](#_Toc36284611)

[**Collegamento ipertestuale** 48](#_Toc36284612)

[**Testo preformattato** 48](#_Toc36284613)

[**Testo deponente** 48](#_Toc36284614)

[**Testo apice** 48](#_Toc36284615)

[**Citazioni** 48](#_Toc36284616)

[**Abbreviazioni** 48](#_Toc36284617)

[**Immagine** 49](#_Toc36284618)

[**Tabelle:** 49](#_Toc36284619)

[**Elenchi puntati:** 49](#_Toc36284620)

[**Elenchi ordinati:** 50](#_Toc36284621)

[**Form** 50](#_Toc36284622)

[**Elementi Form:** 50](#_Toc36284623)

[**Select:** 50](#_Toc36284624)

[**Option:** 50](#_Toc36284625)

[**TextArea:** 50](#_Toc36284626)

[**Input di vario tipo:** 50](#_Toc36284627)

[**Tipologie Input:** 51](#_Toc36284628)

**HTML**

# **Commenti**

<!-- Commento -->

# **Intestazione HTML**

<!DOCTYPE html>

# **Attributi Tag:**

name=”nome identificativo di un singolo elemento usato nei form”

id=”nome identificativo di un singolo elemento”

class=”nome identificativo di N elementi”

style=”proprietà: valore”

# **Sezione HTML**

<html> <!—- Contenuto --> </html>

# **Sezione Head**

<head> <!—- Contenuto --> </head>

# **Importazione di un file (CSS, Font, etc …)**

<link rel=”tipologia del file” type=”tipologia formato” href=”collegamento al file”>

# **Sezione Body**

<body> <!—- Contenuto --> </body>

# **Titolo a grandezza variabile N, da 1 a 6**

<hN>Testo titolo</hN>

# **Paragrafo**

<p>Testo paragrafo</p>

# **Collegamento ipertestuale**

<a href=”link di riferimento”>Testo</a>

# **Testo preformattato**

<pre> <!—- Testo --> </pre>

# **Testo deponente**

<sub> <!—- Testo --> </syb>

# **Testo apice**

<sup> <!—- Testo --> </sup>

# **Citazioni**

<blockquote cite=”link della citazione”>

<!—- Testo Citazione -->

</blockquote>

# **Abbreviazioni**

<abbr title=”testo completo”>

<!—- Testo Abbreviazione -->

</abbr>

# **Immagine**

<img src=”link dell’immagine”>

# **Tabelle:**

<table>

<thead>

<tr>

<th>Colonna 1</th>

<th>Colonna 2</th>

</tr>

</thead>

<tbody>

<tr>

<th>DatoColonna1</th>

<th> DatoColonna2</th>

</tr>

</tbody>

</table>

# **Elenchi puntati:**

<ul>

<li>Elemento1</li>

<li>Elemento2</li>

</ul>

# **Elenchi ordinati:**

<ol>

<li>Elemento1</li>

<li>Elemento2</li>

</ol>

# **Form**

<form>

<!—- Elementi Form -->

</form>

# **Elementi Form:**

## **Select:**

<select>

<!—- Elementi Option -->

</select>

## **Option:**

<option value=”valoreIntrinseco”>

<!—- Valore da mostrare a schermo -->

</select>

## **TextArea:**

<textarea rows=”” cols=””>

<!—- Contenuto pre-esistente (opzionale)-->

</textarea>

## **Input di vario tipo:**

<input type=”tipologiaInput” name=”nomeInput”>

## **Tipologie Input:**

url

button

search

submit

tel

number

range

radio

color

file

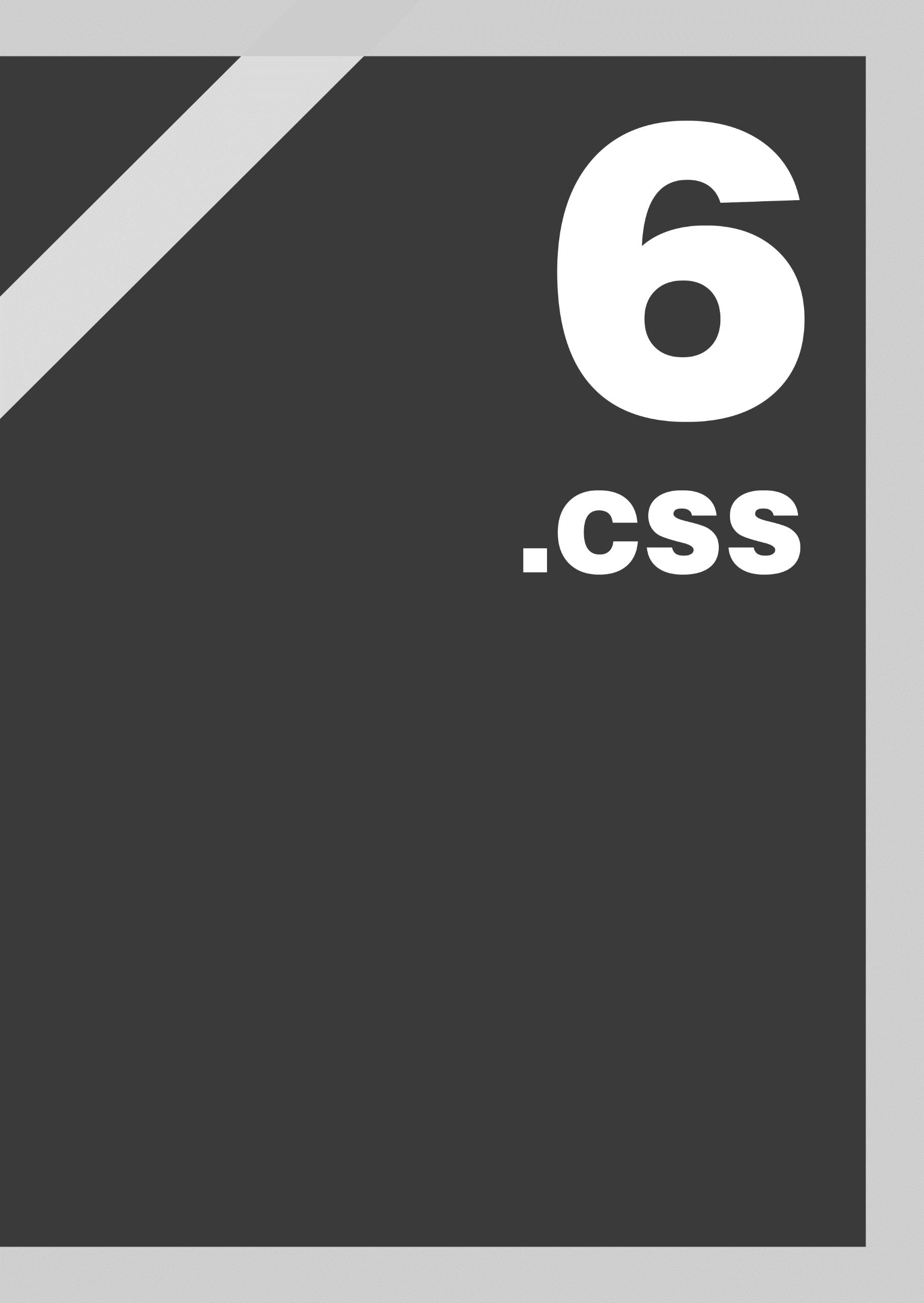
date

checkbox

password

email

text



**Sommario**

[**Commenti** 54](#_Toc36284243)

[**Sintassi** 54](#_Toc36284244)

[**Proprietà** 55](#_Toc36284245)

**CSS**

# **Commenti**

/\* Commento \*/

# **Sintassi**

Tag | .nomeClasse | #nomeID | selector{

proprietà: valore

}

# **Proprietà**

margin-bottom: valore\_numerico px | %

margin-top: valore\_numerico px | %

background-color: rgb(r, g, b) | rgba(r, g, b, a) | “colore\_esadecimale” | hsl(h, s ,l)

background: rgb(r, g, b) | rgba(r, g, b, a) | “colore\_esadecimale” | hsl(h, s ,l) | url(“link immagine”)

color: rgb(r, g, b) | rgba(r, g, b, a) | “colore\_esadecimale” | hsl(h, s ,l)

font-weight: valore\_numerico

font-family: “nome\_font”

visibility: visible | hidden

font-size: valore\_numerico px | pt | %

display: block | inline | flex | table | block-inline | none

text-align: center | left | right

height: valore\_numerico px | em | vh | %

width: valore\_numerico px | em | vw | %

padding: (valore\_numerico px | %) (valore\_numerico px | %) (valore\_numerico px | %) (valore\_numerico px | %)

opacity: valore\_numerico (tra 0 e 1)

padding-right: valore\_numerico px | %

padding-left: valore\_numerico px | %

padding-top: valore\_numerico px | %

padding-bottom: valore\_numerico px | %

margin-right: valore\_numerico px | %

margin-left: valore\_numerico px | %



Sommario

[**Imposta un nome utente Git :** 60](#_Toc36501629)

[**Imposta una e-mail Git :** 60](#_Toc36501630)

[**Creazione di un nuovo repository :** 60](#_Toc36501631)

[**Copia di un repository :** 60](#_Toc36501632)

[**Copia di un repository usando un server remoto :** 60](#_Toc36501633)

[**Visualizzare lista di server remoti salvati con relativo url :** 60](#_Toc36501634)

[**Aggiungere un server remoto :** 60](#_Toc36501635)

[**Aggiungere files della directory del progetto all’index :** 60](#_Toc36501636)

[**Aggiungere files all’index all’head :** 60](#_Toc36501637)

[**Aggiungere files dell’index all’head(senza tracciare il file) :** 61](#_Toc36501638)

[**Annullare un commit :** 61](#_Toc36501639)

[**Cancellare un file da Git :** 61](#_Toc36501640)

[**Ritornare allo stato precedente di un file all’ultimo commit :** 61](#_Toc36501641)

[**Aggiorna il repository locale e il commit più recente :** 61](#_Toc36501642)

[**Upload dei commit nel progetto :** 61](#_Toc36501643)

[**Rinominare un file in remoto :** 61](#_Toc36501644)

[**Eliminare un file in remoto :** 61](#_Toc36501645)

[**Vedere le modifiche del progetto :** 61](#_Toc36501646)

[**Vedere i cambiamenti dei singoli file :** 61](#_Toc36501647)

[**Vedere tutti i commit :** 62](#_Toc36501648)

[**Visualizzare tutte le versioni :** 62](#_Toc36501649)

[**Visualizzare tutte le versioni con un determinato numero :** 62](#_Toc36501650)

[**Creare un tag** 62](#_Toc36501651)

[**Vedere tutte le modifiche di un tag :** 62](#_Toc36501652)

[**Condividere un tag :** 62](#_Toc36501653)

[**Condividere tutti i tag :** 62](#_Toc36501654)

[**Visualizzare la lista dei rami :** 62](#_Toc36501655)

[**Creazione di un Branch :** 62](#_Toc36501656)

[**Cambiare i rami :** 62](#_Toc36501657)

[**Ritornare al branch originale :** 63](#_Toc36501658)

[**Eliminare un ramo :** 63](#_Toc36501659)

[**Creare il ramo e passare a quel branch :** 63](#_Toc36501660)

[**Unire la branch alla repository :** 63](#_Toc36501661)

[**Parametri GIT :** 63](#_Toc36501662)

[**Inizializzare area di lavoro:** 63](#_Toc36501663)

[**Lavorare nel progetto corrente :** 63](#_Toc36501664)

[**Collaborare nel progetto :** 64](#_Toc36501665)

**GIT**

# **Imposta un nome utente Git :**

git config –global user.name *githubName*

# **Imposta una e-mail Git :**

git config –global user.email *email@github.com*

# **Creazione di un nuovo repository :**

git init

# **Copia di un repository :**

git clone */percorso/repository*

# **Copia di un repository usando un server remoto :**

git clone *nomeutente@host:/percorso/repository*

# **Visualizzare lista di server remoti salvati con relativo url :**

git remote -v

# **Aggiungere un server remoto :**

git remote add *identifierRemoteServer urlRemoteServer*

# **Aggiungere files della directory del progetto all’index :**

git add nome file

# **Aggiungere files all’index all’head :**

git commit -m “Messaggio del commit”

# **Aggiungere files dell’index all’head(senza tracciare il file) :**

git commit -a -m “Messaggio del commit”

# **Annullare un commit :**

git commit --amend

# **Cancellare un file da Git :**

git rm *nomeFile*

# **Ritornare allo stato precedente di un file all’ultimo commit :**

git checkout --*nomeFile*

# **Aggiorna il repository locale e il commit più recente :**

git pull

# **Upload dei commit nel progetto :**

git pudh identifierRemoteServer branchName

# **Rinominare un file in remoto :**

git remote idRemoteServer oldFile newFile

# **Eliminare un file in remoto :**

git remote rm nameFile

# **Vedere le modifiche del progetto :**

git status

# **Vedere i cambiamenti dei singoli file :**

git diff

# **Vedere tutti i commit :**

git log

# **Visualizzare tutte le versioni :**

git tag

# **Visualizzare tutte le versioni con un determinato numero :**

git tag -l number\*

# **Creare un tag**

git tag -a SWversion -m “ nota ”

# **Vedere tutte le modifiche di un tag :**

git show 1.2.3rc1

# **Condividere un tag :**

git push IdentifierRemoteServer tagDaPubblicare

# **Condividere tutti i tag :**

git push IdentifierRemoteServer --tag

# **Visualizzare la lista dei rami :**

git branch

# **Creazione di un Branch :**

git branch branchName

# **Cambiare i rami :**

git checkout branchName

# **Ritornare al branch originale :**

git checkout master

# **Eliminare un ramo :**

git branch -d branchName

# **Creare il ramo e passare a quel branch :**

git branch -b branchName

# **Unire la branch alla repository :**

git checkout master

git merge branchName

# **Parametri GIT :**

# **Inizializzare area di lavoro:**

**Clone:** *Clona un repository in una cartella*

**Init:** *Crea un git repository o ne inizializza uno*

# **Lavorare nel progetto corrente :**

**Add:** *Aggiungere i file nell’index*

**Mv:** *Muove o rinomina un file o una directory*

**Reset:** *Resetta il corrente Head nello stato specificato*

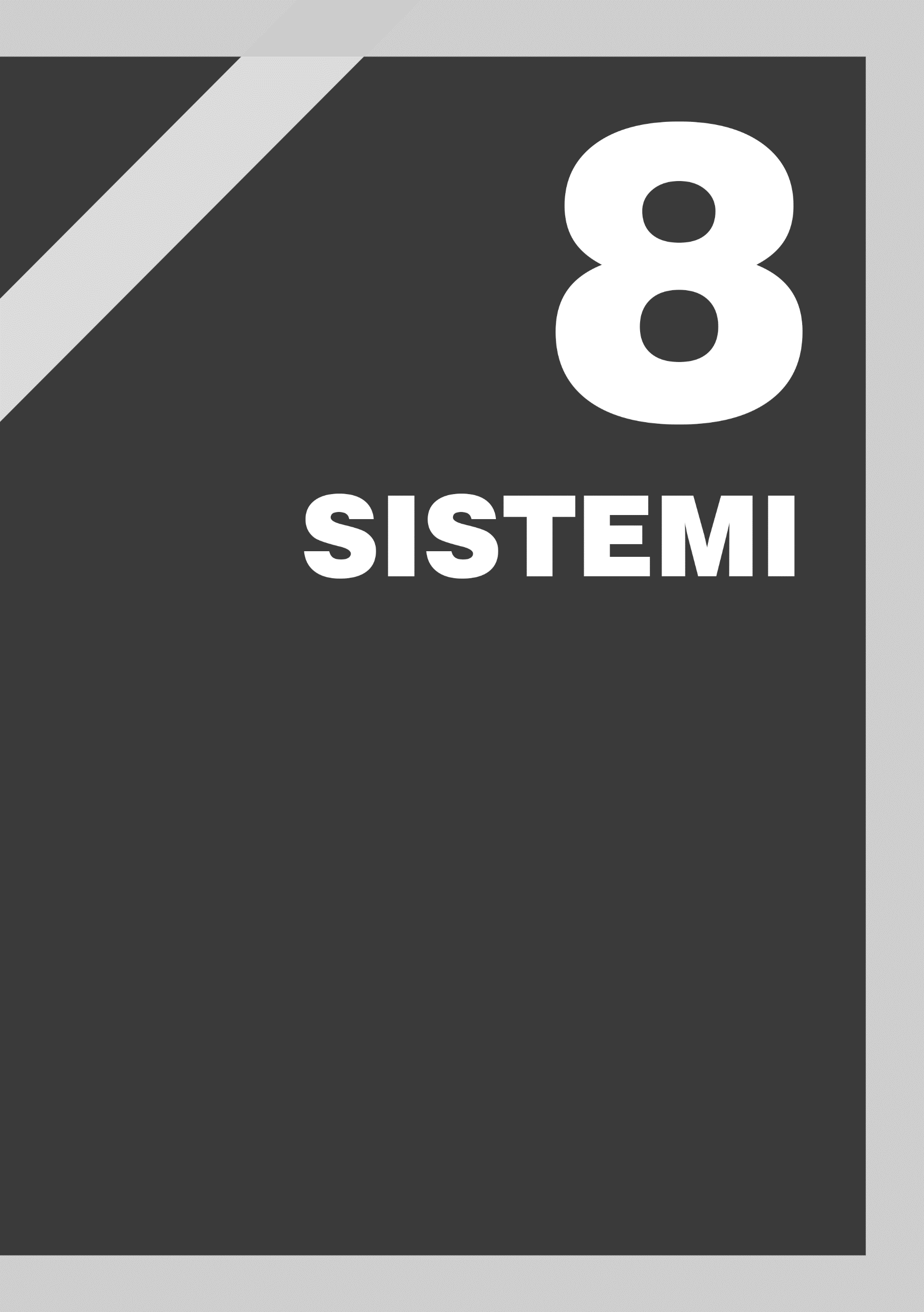
**RM:** *Rimuove il file dalla directory corrente e nell’index*

# **Collaborare nel progetto :**

**Fetch:** *Scarica oggetti e riferimenti da un altro repository*

**Pull:** *Recupera e integra con un altro repository o un branch*

**Push:** *Aggiorna i riferimenti insieme agli oggetti assocciati*



**Sommario**

[**FTP Server su Ubuntu 18** 67](#_Toc36285752)

[**LAMP Server (Apache, MySql, PHP) su Ubuntu 18** 67](#_Toc36285753)

[**Samba Server su Ubuntu 18** 69](#_Toc36285754)

[**Mail Server su Ubuntu 18** 71](#_Toc36285755)

[**OpenVPN Server su Ubuntu 18** 72](#_Toc36285756)

[**Mail Server su Ubuntu 18** 73](#_Toc36285757)

[**Server DHCP su ubuntu 18** 74](#_Toc36285758)

[**Server DHCP su ubuntu 18** 75](#_Toc36285759)

[**Server SSH su ubuntu 18** 76](#_Toc36285760)

[**Server SSH su ubuntu 18** 77](#_Toc36285761)

[**CLI Cisco** 78](#_Toc36285762)

[**Privilegi root :** 78](#_Toc36285763)

[**Configurazione globale:** 78](#_Toc36285764)

[**configurazione interfaccia :** 78](#_Toc36285765)

[**Assegnamento indirizzo IP :** 78](#_Toc36285766)

[**Mantenimento dell’interfaccia sempre accesa :** 78](#_Toc36285767)

[**Impostazione di una rotta statica su router :** 78](#_Toc36285768)

[**Impostazione di una rotta statica su router :** 78](#_Toc36285769)

[**Visualizzazione delle rotte su router :** 78](#_Toc36285770)

[**Visualizzazione delle interfacce di rete su router :** 78](#_Toc36285771)

[**Impostazione ACL :** 79](#_Toc36285772)

[**Eliminazione ACL :** 79](#_Toc36285773)

[**Impostazione ACL extended :** 79](#_Toc36285774)

[**Visualizzazione ACL :** 79](#_Toc36285775)

[**Creazione VLAN :** 79](#_Toc36285776)

[**Assegnazione VLAN ad una porta :** 79](#_Toc36285777)

[**Impostazione porta in trunk :** 79](#_Toc36285778)

[**Impostazione subinterface:** 80](#_Toc36285779)

[**Ritorno ad una precedente modalità:** 80](#_Toc36285780)

[**Ritorno alla modalità enable da qualsiasi modalità:** 80](#_Toc36285781)

**Sistemi e Reti :**

# **FTP Server su Ubuntu 18**

**Aggiornarmento Server :**

sudo apt-get update

**Installazione servizio vsftpd :**

sudo apt-get install vsftpd

**Controllo funzionamento :**

sudo service vsftpd status

# **LAMP Server (Apache, MySql, PHP) su Ubuntu 18**

**Installazione Apache :**

sudo apt install -y apache2 apache2-utils

**Controllo servizio :**

sudo systemctl status apache2

**In caso di blocco da parte del firewall :**

sudo iptables -I INPUT -p tcp –dport 80 -j ACCEPT

**E di seguito eseguire :**

sudo ufw allow http

**Impostazione dello user Apache per la possessione delle directory delle pagine Web Apache :**

sudo chown www-data:www-data /var/www/html -R

**Installazione MariaDB(MySql) :**

sudo apt install mariadb-server mariadb-client

**Script di sicurezza post installazione :**

Sudo mysql\_secure\_installation

**Installazione PHP 7.2 :**

sudo apt install php7.2 libapache2-mod-php7.2 php7.2-msyql php-common php7.2-cli php7.2-common php7.2-json php7.2-opcache php7.2-readline

**Attivazione modulo apache php7.2 :**

sudo a2enmod php7.2

**Riavvio apache2 :**

sudo systemctl restart apache2

# **Samba Server su Ubuntu 18**

**Aggiornarmento Server :**

sudo apt-get update

**Installazione servizio :**

sudo apt install samba samba-common-bin

**Controllo servizio :**

sudo service smbd status

**Creazione di una condivisione privata di Samba. Modifica del file /etc/samba/smb.conf :**

sudo nano /etc/samba/smb.conf

**Creazione utente Linux per Samba :**

sudo add user *username*

**Creazione password per Samba :**

sudo smbpasswd -a *username*

**Creazione gruppo Samba :**

sudo groupadd samba

**Aggiunta dell’utente al gruppo Samba :**

sudo gpasswd -a username samba

**Creazione cartella privata per la condivisione :**

sudo mkdir /srv/private/

**Permessi di lettura|scrittura|esecuzione della cartella condivisa per il gruppo Samba :**

sudo setfacl -R -m “g:samba:rwx” /srv/private

**Controllo dei possibili errori :**

sudo testparm

**Riavvio del servizio :**

sudo systemctl restart smbd

**Creazione di una condivisione pubblica di Samba. Modifica del file /etc/samba/smb.conf :**

sudo nano /etc/samba/smb.conf

**Creazione cartella pubblica per la condivisione :**

sudo mkdir /srv/public/

**Permessi per l’utente anonimo:**

sudo setfacl -R -m “g:nobody:rwx” /srv/private

**Riavvio del servizio :**

sudo systemctl restart smbd

# **Mail Server su Ubuntu 18**

**Aggiornarmento Server :**

sudo apt-get update

**Installazione Postfix :**

sudo apt install postfix

**Backup del file configurazione principale :**

sudo cp /etc/postfix/main.cf /etc/postfix/main.cf.backup

**Modifica file /etc/postfix/main.cf :**

sudo nano /etc/postfix/main.cf

**Abilitazione servizio :**

sudo systemctl enable postfix

**Accensione servizio :**

sudo systemctl start postfix

**Configurazione firewall :**

sudo ufw allow “Postfix”

sudo ufw allow “Postfix SMTPS”

sudo ufw allow “Postfix Submission”

# **OpenVPN Server su Ubuntu 18**

**Aggiornarmento Server :**

sudo apt-get update

**Scaricamento a avvio di openvpn-install.sh :**

sudo wget <https://git.io/vpn> -O openvpn-install.sh

**Permessi :**

sudo chmod +x openvpn-install.sh

**Avvio openvpn-install.sh :**

sudo ./openvpn-install.sh

**Riavvio e controllo servizio :**

sudo systemctl stop openvpn@server

sudo systemctl start openvpn@server

sudo systemctl restart openvpn@server

sudo systemctl statys openvpn@server

# **Mail Server su Ubuntu 18**

**Aggiornarmento Server :**

sudo apt-get update

**Installazione Squid :**

sudo apt install squid

**Backup del file di configurazione :**

sudo cp /etc/squid/squid.conf /etc/squid/squid.conf.backup

**Modifica file /etc/squid/squid.conf :**

sudo nano /etc/squid/squid.conf

**Modifica file ban\_domains.txt :**

sudo nano /etc/squid/ban\_domains.txt

**Riavvio servizio :**

sudo systemctl restart squid

# **Server DHCP su ubuntu 18**

**Aggiornarmento Server :**

sudo apt-get update

**Installazione Server DHCP :**

sudo apt install isc-dhcp-server

**Modifica file per impostazione scheda di rete :**

sudo nano /etc/default/isc-dhcp-server

**Configurazione del server tramite file :**

sudo nano/etc/dhcp/dhcpd.conf

**Avvio servizio :**

sudo systemctl start isc-dhcp-server

**Controllo servizio :**

sudo systemctl status isc-dhcp-server

# **Server DHCP su ubuntu 18**

**Aggiornarmento Server :**

sudo apt-get update

**Installazione server DNS :**

sudo apt install bind9 bind9utils bind9-doc dnsutils

**Modifica file forward copiato da /etc/bind/named.conf.local :**

sudo nano /etc/bind/fwd.*cognome*.tes.mi.it

**Modifica file reverse copiato da /etc/bind/db.127 :**

sudo nano /etc/bind/rvs.*cognome*.tes.mi.it

**Abilitazione servizio:**

sudo systemctl enable bind9

**Avvio servizio:**

sudo systemctl start bind9

**Controllo servizio :**

sudo systemctl status bind9

# **Server SSH su ubuntu 18**

**Aggiornarmento Server :**

sudo apt-get update

**Installazione Server OpenSSH :**

sudo apt install openssh-server

**Abilitazione servizio:**

sudo systemctl enable ssh

**Avvio servizio:**

sudo systemctl start ssh

**Controllo servizio :**

sudo systemctl status ssh

# **Server SSH su ubuntu 18**

**Aggiornarmento Server :**

sudo apt-get update

**Installazione Server Radius :**

sudo apt install freeradius

**Modifica file per la configurazione dei client :**

sudo vim /etc/freeradius/3.0/client.conf

**Modifica file per la configurazione degli users:**

sudo vim /etc/freeradius/3.0/users

**Abilitazione servizio:**

sudo systemctl enable freeradius

**Avvio servizio:**

sudo systemctl start freeradius

**Controllo servizio :**

sudo systemctl status freeradius

# **CLI Cisco**

## **Privilegi root :**

enable

## **Configurazione globale:**

configure terminal

## **Configurazione interfaccia :**

interface *interfaccia numeroInterfaccia*

## **Assegnamento indirizzo IP :**

ip address *indirizzoIP maschera*

## **Mantenimento dell’interfaccia sempre accesa :**

no shutdown

## **Impostazione di una rotta statica su router :**

ip route *rete maschera nexthop*

## **Impostazione di una rotta statica su router :**

no ip route *rete maschera nexthop*

## **Visualizzazione delle rotte su router :**

show ip route

## **Visualizzazione delle interfacce di rete su router :**

show interfaces

## **Impostazione ACL :**

access-list *numero permit|deny ip maschera*

numer

## **Eliminazione ACL :**

no access-list *numero*

## **Impostazione ACL extended :**

access-list *numero permit|deny protocollo ipInizio mascheraInizio ipFine mascheraFine numeroPorta*

## **Visualizzazione ACL :**

show access-lists

## **Creazione VLAN :**

vlan *numeroVlan*

## **Assegnazione VLAN ad una porta :**

interface *porta*

switchport mode access

switchport access vlan *numeroVlan*

## **Impostazione porta in trunk :**

interface *porta*

switchport mode trunk

switch trunk allowed vlan add *numeroVlan*

switchport trunk allowed vlan add *numeroVlanSwitchDiverso*

## **Impostazione subinterface:**

Interface nomeInterfaccia numeroInterfaccia.numerosottoInterfaccia

## **Ritorno ad una precedente modalità:**

exit

## **Ritorno alla modalità enable da qualsiasi modalità:**

end