

HTML (HyperText Markup Language)

ITCG Fermi

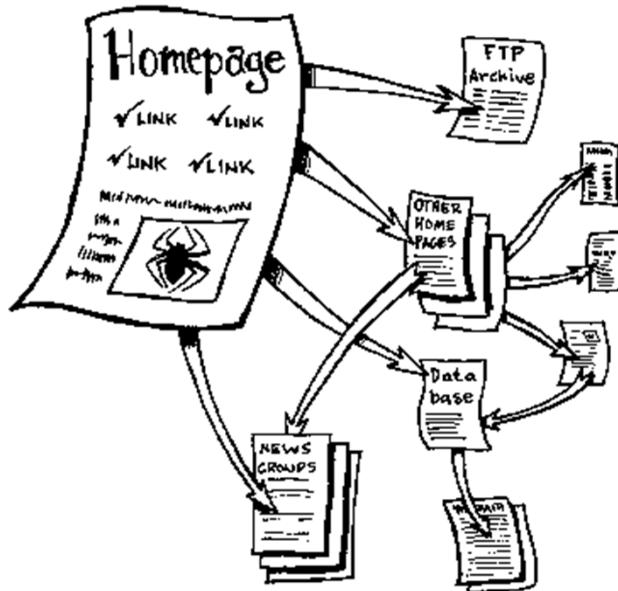
Prof. Montemurro

Linguaggi per Realizzare Pagine Web

1. Il linguaggio **HTML (HyperText Markup Language, o linguaggio di contrassegno per gli ipertesti)** consente la creazione di pagine web attraverso **file di testo** che sono interpretati dal browser che le visualizza in **forma grafica**. È un linguaggio per la gestione di ipertesti; **non** è un linguaggio di programmazione in quanto non viene utilizzato per svolgere elaborazioni, ma è un **linguaggio di formattazione della pagina**.
Ipertesto: testo multidimensionale (iper-), cioè a più dimensioni, infatti è costituito da un insieme di documenti che hanno tra loro un **nesso logico** in quanto si riferiscono ad una stessa classe di argomenti che possono essere consultati in modo **non lineare** mediante link a documenti esterni (**hyperlink**) per associazione di idee o di termini.
Navigazione ipertestuale: modalità di lettura dei documenti che prevede il salto da uno all'altro tramite i link.

Prof. Montemurro

Linguaggi per Realizzare Pagine Web



Prof. Montemurro

Traduttori: Compilatori ed Interpreti

- 1. Compilatori:** programmi che prendono in ingresso l'intero programma (file sorgente/i), e che restituiscono in uscita la rappresentazione dell'intero programma in linguaggio macchina (o codice binario 01...).
- 2. Interpreti:** programmi che traducono (in linguaggio macchina) ed eseguono direttamente ciascuna istruzione del programma/script scritto nel file sorgente/i, istruzione per istruzione.

Prof. Montemurro

Linguaggi per Realizzare Pagine Web

2. Il linguaggio **JavaScript JS** è:

- i. un linguaggio interpretato in quanto il codice non viene compilato, ma viene eseguito istruzione per istruzione direttamente dal browser;
- ii. un linguaggio di scripting (quindi è un linguaggio di programmazione) per cui è meglio parlare di script e non di programma.

Script: testo contenente una sequenza di istruzioni che può essere innestata anche nel codice di altri linguaggi, ed è eseguita da un programma interprete.

Limite dell'HTML e di JavaScript: non è possibile realizzare pagine web che consentano all'utente di accedere ai dati contenuti in un database che si trova su un server web.

Prof. Montemurro

Linguaggi per Realizzare Pagine Web

Linguaggi lato client (HTML, JavaScript ...): interpretazione delle istruzioni viene fatta direttamente dal browser web nel computer dell'utente che svolge il ruolo di client (richiedente).

Linguaggi lato server (PHP ...): interpretazione delle istruzioni avviene sul server, ed il risultato dell'elaborazione viene inviato al browser; le pagine web (dinamiche) contengono quindi il risultato del codice eseguito sul server.

Pagina web dinamica: pagina web che viene costruita dal server quando l'utente la richiede. A differenza di una pagina statica, che è un file HTML predefinito e identico per tutti gli utenti, una pagina dinamica può variare in base a diversi fattori, quindi è interattiva e personalizzabile.

Prof. Montemurro

Linguaggi per Realizzare Pagine Web

Server web (es. Apache): software per la gestione dei servizi web di un computer host o di un server di rete; tali servizi si basano sui protocolli standard delle reti e di internet, in particolare sul protocollo HTTP (HyperText Transfer Protocol) che è il principale protocollo informatico che consente il trasferimento di dati da e verso pagine web.

Il **PHP** (acronimo ricorsivo **PHP Hypertext Preprocessor, o preprocessore di ipertesti PHP**) è un linguaggio che estende le funzionalità del server web in quanto consente l'interpretazione di file con estensione *.php* contenenti il codice dell'applicazione lato server, oltre che naturalmente l'interpretazione dei classici marcatori (o tag) dei file con estensione *.html*.

Prof. Montemurro

Linguaggi per Realizzare Pagine Web

Il PHP è:

1. un linguaggio interpretato in quanto il codice non viene compilato, ma viene eseguito istruzione per istruzione direttamente dal server; l'**interprete PHP** viene aggiunto al server web come modulo esterno;
2. un linguaggio di scripting (quindi è un linguaggio di programmazione) in quanto permette di innestare gli script all'interno delle pagine web dove sono presenti i marcatori del linguaggio HTML. In questo modo si possono realizzare in modo dinamico **pagine web lato server** che non dipendono solo dai marcatori statici del linguaggio HTML, ma anche dalle azioni degli script realizzati tramite il linguaggio PHP;
3. un linguaggio integrato nel linguaggio HTML.

Prof. Montemurro

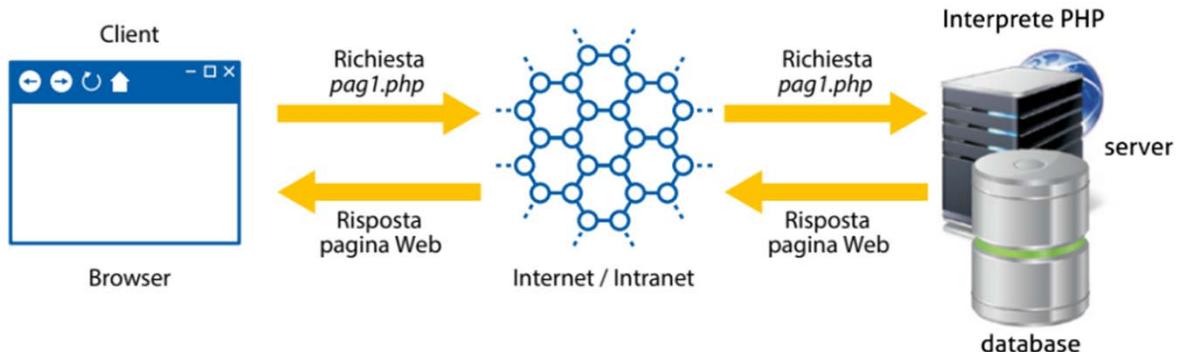
Linguaggi per Realizzare Pagine Web

Fasi eseguite dal server web quando viene richiesta una pagina con estensione `.php`:

1. legge il file di testo `.php` riga per riga;
2. quando trova i marcatori HTML li spedisce al browser;
3. quando trova i blocchi di codice HTML:
 - i. ne esegue l'interpretazione;
 - ii. recupera gli eventuali dati richiesti prelevandoli dai file o dai database del server;
 - iii. restituisce una pagina web, creata quindi in modo dinamico, visualizzabile dal browser.

Prof. Montemurro

Linguaggi per Realizzare Pagine Web



Prof. Montemurro

Caratteristiche delle Pagine PHP

Pagina con codice PHP

```
<HTML>
  - -
<BODY>
  <?php
    //codice PHP
  ?>
</BODY>
</HTML>
```

Pagina ricevuta dal browser

```
<HTML>
  - -
<BODY>
  solo HTML
</BODY>
</HTML>
```

Interprete PHP

Prof. Montemurro

HTML

Il linguaggio HTML è diventato lo standard nell'architettura WWW per creare e riconoscere i documenti ipermediali (o ipertesti multimediali). Le specifiche del linguaggio sono stabilite da **W3C (World Wide Web Consortium)** il quale è l'ente che regola gli standard del web (consultabili alla pagina w3.org).

Documento in formato web: file di testo progettato per essere visualizzato su un browser web. Questi documenti sono scritti utilizzando linguaggi specifici e strutture che consentono loro di essere resi e interpretati correttamente in un ambiente online. Tali documenti possono essere utilizzati offline (es. file HTML salvato localmente), e possono esistere indipendentemente dai protocolli di rete (HTTP/HTTPS).

Prof. Montemurro

HTML

Documento WWW: documento accessibile attraverso il World Wide Web, quindi è un particolare tipo di documento in formato web. Tale documento:

1. deve essere ospitato su un server web e raggiungibile tramite un URL (Uniform Resource Locator) usando il protocollo HTTP o HTTPS; quindi deve essere pubblicato e deve essere accessibile sul web;
2. è solitamente un documento HTML (spesso accompagnato da CSS e JavaScript) che costituisce una pagina web visualizzabile da un utente tramite un browser.

I documenti in formato web, essendo dei normali file di testo, possono essere aperti e modificati con un qualunque editor di testi (es. Blocco Note, Microsoft Word).

Prof. Montemurro

HTML

Quando si apre un documento in formato web con un browser, sullo schermo viene visualizzata una pagina in formato grafico: il browser elabora i documenti in formato web, in particolare trasforma i codici scritti in tali documenti in comandi per la costruzione della pagina.

I file scritti in HTML hanno estensione **.html** oppure **.htm**; il suffisso non è obbligatorio, ma è utilizzato normalmente per identificare il tipo di file.

Prof. Montemurro

Marcatori (o Tag) HTML

Tag (o marcatori) HTML: codici di formazione della pagina, sequenze di caratteri proprie del linguaggio HTML che non fanno parte del testo normale, e che sono comprese tra i simboli di minore < e maggiore >. Tali sequenze consentono (1) di realizzare gli elementi caratteristici dell'ipertesto (link, oggetti grafici e multimediali), e permettono (2) di semplificare l'impaginazione dei documenti complessi (specificando stili del testo, titoli, paragrafi, liste).

Prof. Montemurro

Marcatori (o Tag) HTML

In genere ad ogni **tag (di apertura)** corrisponde un **tag di chiusura** il quale è un tag con lo stesso nome del tag di apertura preceduto dalla barra / . L'azione di un tag ha effetto sul testo compreso tra il tag di apertura ed il tag di chiusura:

```
<comando parametro1 parametro2> ... </comando>
```

Esempio: il tag di apertura **** attiva il grassetto nella scrittura dei caratteri, mentre il tag di chiusura **** indica la fine dell'azione. Dunque, tutti i caratteri compresi tra **** e **** saranno visualizzati in grassetto dal browser.

Cognome: Rossi **Nome:** Giovanni

Cognome: Rossi **Nome:** Giovanni

Prof. Montemurro

Marcatori (o Tag) HTML

Convenzione: i tag HTML devono essere scritti in minuscolo (funziona anche se si scrivono in maiuscolo). Fa eccezione la parola chiave **DOCTYPE** la quale si trova nella prima riga di ogni documento HTML, e serve per specificare che si sta usando il linguaggio HTML (così che il browser lo sappia).

```
<!DOCTYPE html>
```

Prof. Montemurro

Struttura di una Pagina HTML

Di norma un testo HTML è suddiviso in almeno due sezioni:

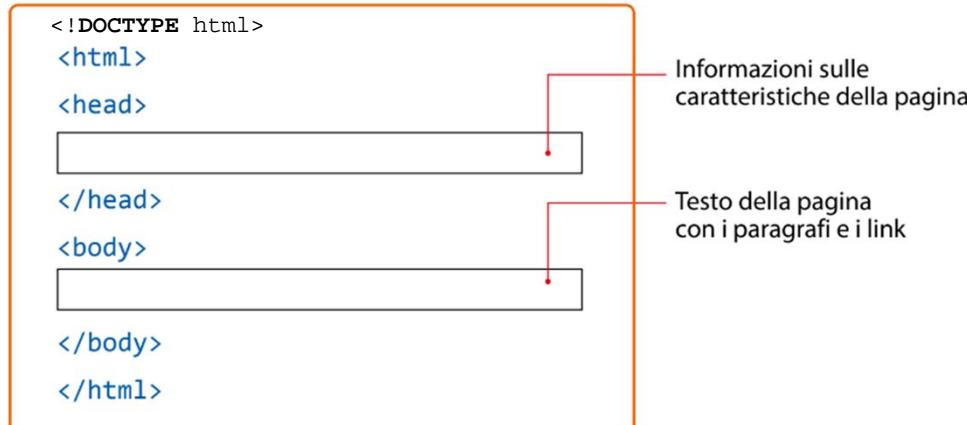
1. l'**intestazione** (o **head**) la quale contiene informazioni sul documento (**metadati**: dati sui dati), e viene codificata dai tag **<head>** e **</head>** ;
2. il **corpo** (o **body**) il quale contiene il documento vero e proprio con tutti gli elementi caratteristici di un ipertesto, e viene codificato dai tag **<body>** e **</body>** .

Inoltre, un documento HTML è di norma delimitato dai tag **<html>** e **</html>**, perché alcuni browser devono essere informati sulla porzione di testo che deve essere interpretata come testo HTML. Per facilitare il lavoro dei motori di ricerca, può essere utile aggiungere l'attributo **lang** all'interno del tag **<html>**; tale attributo serve per specificare la lingua della pagina web:

```
<html lang="it">
```

Prof. Montemurro

Sezioni di una Pagina HTML



Prof. Montemurro

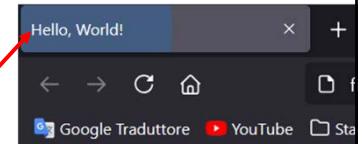
Intestazione con Titolo e Metadati

Nella sezione di intestazione `<html>` viene descritta la pagina (titolo, descrizione, parole chiave, autore, codifica della pagina ecc.).

Titolo di una pagina HTML: ogni pagina HTML ne deve avere uno, viene usato per identificare il documento (es. quando si cerca un documento tra tanti attraverso un motore di ricerca, o quando una pagina viene inserita nei siti preferiti). E' opportuno quindi che (1) ogni titolo sia diverso dai titoli delle altre pagine che formano il sito, e che (2) ogni titolo sia formato da una frase significativa che rimandi al contenuto della pagina.

Il titolo viene racchiuso tra i tag `<title>` e `</title>` ed è il nome della scheda (tab) della finestra del browser relativa alla pagina HTML aperta.

`<title>Hello World!</title>`



Hello, World!

Intestazione con Titolo e Metadati

Meta tag: sono i tag che contengono i **metadati** della pagina (informazioni sul contenuto della pagina, le parole chiave, l'autore, la codifica del testo). Tali tag sono inseriti nella sezione di intestazione della pagina HTML, tra <head> e </head>.

I meta tag sono identificati dal tag <**meta**>, e la struttura generale della frase è:

```
<meta name="..." content="...">
```

Valori più usati dell'attributo name:

1. **description** per la descrizione della pagina HTML;
2. **keywords** per specificare le parole chiave contenute nella pagina HTML;
3. **author** per specificare l'autore della pagina HTML.

Prof. Montemurro

Intestazione con Titolo e Metadati

Esempio: pagina web di un sito web scolastico.

```
<meta name="description" content="Sito Web dell'ITCG  
Fermi">  
<meta name="keywords" lang="it" content="scuola,  
Fermi, Istituto, Informatica, Web, Tecnologie,  
studenti, docenti, genitori">  
<meta name="author" content="Webmaster, redazione">
```

Prof. Montemurro

Intestazione con Titolo e Metadati

Codifica della pagina: se non si indica la codifica corretta, può accadere che alcuni caratteri non standard, per esempio le lettere accentate, non vengano visualizzati correttamente. Per specificare la codifica **UTF-8**, che contiene tutti i caratteri dell'Europa occidentale, si utilizza il codice:

```
<meta charset="UTF-8">
```

Prof. Montemurro

Intestazione con Titolo e Metadati

Modello di base di una pagina HTML, completo di titolo e codifica UTF-8.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Titolo della pagina</title>
  </head>
  <body>
    Contenuto della pagina
  </body>
</html>
```

Prof. Montemurro

Commenti in HTML

```
<!-- commento su una riga -->
```

```
<!-- commento su  
più righe -->
```

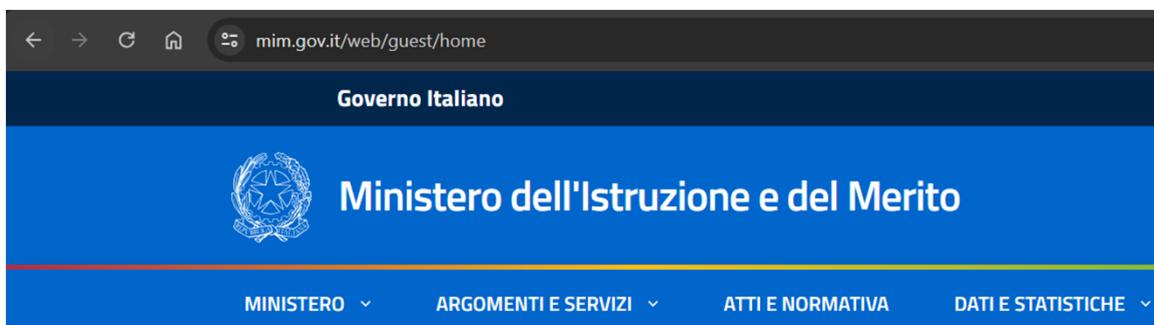
Notare che il punto esclamativo compare solo all'inizio.

I commenti possono essere utili per nascondere temporaneamente la visualizzazione di una parte della pagina.

Prof. Montemurro

Visualizzare Codice HTML di una Pagina Web

Nel browser utilizzare la scorciatoia da tastiera **Ctrl + U** per visualizzare il codice HTML sorgente di una pagina web.



Prof. Montemurro

Visualizzare Codice HTML di una Pagina Web

Nel browser utilizzare la scorciatoia da tastiera **Ctrl + U** per visualizzare il codice HTML sorgente di una pagina web.

```
<!DOCTYPE html>
<html class="ltr no-js" dir="ltr" lang="it-IT">
<head>
    <meta charset="UTF-8">
</head>
<body>
    <div class="body_wrapper push_container" id="page_top">
</body>
</html>
```

Prof. Montemurro

Formattazione di una Pagina Web con HTML

Anche se la soluzione più efficace per formattare le pagine Web si basa sull'uso dei fogli di stile, il linguaggio HTML comprende alcuni **tag stilistici** che permettono di cambiare lo stile del testo e della pagina.

Dimensione dei caratteri: per distinguere il titolo dal resto del testo, per definire i sottotitoli e per evidenziare alcune parti del testo. Esistono sei valori di dimensioni dei caratteri ($n = 1, 2, 3, 4, 5, 6$ dove 1 è relativo alla dimensione più grande), questi valori sono preceduti nel tag dalla lettera **h** (**heading**):

`<hn>...</hn>`

Prof. Montemurro

Formattazione di una Pagina Web con HTML

Esempio: codice HTML con varie dimensioni dei caratteri.

```
<h1>Mobile marketing e social marketing</h1>
<h2>Strategie promozionali delle aziende</h2>
<h3>Tecnologie Web e informatica mobile</h3>
```

Nel browser si ottiene:

Mobile marketing e social marketing

Strategie promozionali delle aziende

Tecnologie Web e informatica mobile

Prof. Montemurro

Formattazione di una Pagina Web con HTML

Stile del testo: alcune parti di testo possono essere evidenziate usando uno stile diverso dal testo normale, come il corsivo, il neretto o il sottolineato. La parte di testo deve essere racchiusa tra una delle seguenti coppie di tag:

- **<i>...</i>** per ottenere il testo in **corsivo** ;
- **...** per il **neretto** (o **grassetto**);
- **<u>...</u>** per il **sottolineato**.

Osservazione: i sta per italic, b per bold, e u per underline.

Prof. Montemurro

Formattazione di una Pagina Web con HTML

Le tecnologie per l'informatica mobile e le applicazioni del **Web 2.0** hanno consentito alle aziende di sviluppare forme di "**marketing non convenzionale**" attuato mediante strategie promozionali che sfruttano metodi di comunicazione, con i potenziali clienti, più efficaci rispetto alla classica attività pubblicitaria. Questa è l'evoluzione del marketing che si presenta con due aspetti: **mobile marketing**, realizzato attraverso specifiche app per **smartphone** e **tablet**, e **social marketing**, basato sull'uso dei **social network**.

Prof. Montemurro

Formattazione di una Pagina Web con HTML

Nel browser si ottiene:

Le tecnologie per l'informatica mobile e le applicazioni del Web 2.0 hanno consentito alle aziende di sviluppare forme di "*marketing non convenzionale*" attuato mediante strategie promozionali che sfruttano metodi di comunicazione, con i potenziali clienti, più efficaci rispetto alla classica attività pubblicitaria. Questa è l'evoluzione del marketing che si presenta con due aspetti: **mobile marketing**, realizzato attraverso specifiche app per *smartphone* e *tablet*, e **social marketing**, basato sull'uso dei *social network*.

Prof. Montemurro

Formattazione di una Pagina Web con HTML

Paragrafo: porzione di testo separata dalle altre da una riga vuota.

1. C'è una riga vuota tra ciò che sta prima del paragrafo e l'inizio del paragrafo stesso.
2. C'è una riga vuota tra la fine del paragrafo e ciò che sta dopo il paragrafo stesso.

I tag per creare un paragrafo sono `<p>...</p>`; questi sono usati anche per impostare aspetti tipografici (colore, tipo di font, dimensione del font ecc.) (si fa coi fogli di stile CSS).

Ritorno a capo: tag `
` (o `
`).

Linea orizzontale: tag `<hr>` (o `<hr/>`).

I tag `
` e `<hr>` sono chiamati **tag vuoti** (o **empty tag**) perché non hanno il corrispondente tag di chiusura.

Prof. Montemurro

Formattazione di una Pagina Web con HTML

Esempio: riga vuota prima e dopo il paragrafo col tag p

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Hello, World!</title>
  </head>
  <body>
    Testo prima del paragrafo. <br>
    Testo prima del paragrafo con ritorno a capo.
    <p>
      Questo è un paragrafo.
    </p>
    Testo dopo il paragrafo.
  </body>
</html>
```

Testo prima del paragrafo.
Testo prima del paragrafo con ritorno a capo.
Questo è un paragrafo.
Testo dopo il paragrafo.

Prof. Montemurro

Formattazione di una Pagina Web con HTML

Esempio: tag hr

```
<!DOCTYPE html>
<html>
  <body>
    <p>Questo è il primo paragrafo.</p>
    <hr>
    <p>Questo è il secondo.</p>
  </body>
</html>
```

Nel browser si ottiene:

Questo è il primo paragrafo.

Questo è il secondo.

Prof. Montemurro

Formattazione di una Pagina Web con HTML

Ritorno a capo automatico (word-wrapping): spezzare le parole lunghe andando a capo alla riga successiva.

Testo preformattato: HTML adotta il ritorno a capo automatico (word-wrapping) come impostazione predefinita. Per evitare questa operazione occorre racchiudere la parte di testo tra la coppia di tag indicanti il **testo preformattato** (es con Microsoft Word):

`<pre>...</pre>`

in questo modo si possono mantenere il formato originale (spazi, tabulazioni, rientri) di un testo creato con un word processor.

Il tag `<pre>`, tuttavia, è da considerarsi obsoleto per cui è opportuno dare la preferenza alle impostazioni tramite i fogli di stile CSS.

Prof. Montemurro

Formattazione di una Pagina Web con HTML

Esempio: tag pre

```
<pre>
  Hello World,
  this text is inside a pre tag, all white   spaces   are
  preserved
</pre>
```

Nel browser si ottiene:

```
Hello World,
this text is inside a pre tag, all white   spaces   are
preserved
```

Prof. Montemurro

Formattazione di una Pagina Web con HTML

In HTML l'**indentazione tramite spazi non** funziona: più spazi di seguito vengono raggruppati in un unico spazio.

Esempio

```
<body>
    ciao      Pippo
</body>
```

Nel browser si ottiene:

ciao Pippo

Prof. Montemurro

Formattazione di una Pagina Web con HTML

Colori: ...

Premessa: basi di numerazione

1. **Base decimale (o base 10):** 0123456789
2. **Base esadecimale (o base 16):** 0123456789ABCDEF

Esempio

Base 10: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 ...

Base 16: 0 1 2 3 4 5 6 7 8 9 A B C D E F 10 ...

Base 10	Base 16	Base 2
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Prof. Montemurro

Formattazione di una Pagina Web con HTML

Utilità della base 16: esistenza di una relazione coi numeri binari che ci permette di esprimere i numeri esadecimali in forma compatta tramite i numeri esadecimali; **ogni numero esadecimale singolo (cioè quelli tra 0 ed F), corrisponde a 4 bit, cioè a 4 cifre binarie.**

Esempio

Numero binario: 100100001011101010

Pezzi da 4 bit: 0010 0100 0010 1110 1010

Numero esadecimale (guarda tabella): 242EA

Base 10	Base 16	Base 2
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Prof. Montemurro

Formattazione di una Pagina Web con HTML

Colori: per rappresentare i colori dei caratteri o dello sfondo di una pagina, il linguaggio HTML utilizza una combinazione dei tre colori fondamentali **RGB** (**Red**, **Green**, **Blue**) attraverso tre numeri:

1. il primo numero è relativo all'intensità del rosso;
2. il secondo numero è relativo all'intensità del verde;
3. il terzo numero è relativo all'intensità del blu.

Ciascun numero è compreso tra 0 e 255 (se lo esprimiamo in base 10) o, equivalentemente, tra 00 e FF (se lo esprimiamo in base 16). Nel codice HTML i tre numeri devono essere espressi in base 16, dunque ogni colore è espresso da sei cifre esadecimali (due per ciascuno dei tre colori: **RRGGBB**).

Prof. Montemurro

Formattazione di una Pagina Web con HTML

Esempi: colori nero, bianco, colori puri (**rosso**, **verde**, **blu**), colori intermedi

Codice	Colore	Spiegazione
000000	Nero	Livello di rosso, di verde e di blu a zero; il nero equivale a nessun colore.
FFFFFF	Bianco	Livello di rosso, di verde e di blu al valore massimo; il bianco equivale alla somma di tutti i colori.
FF0000	Rosso	Livello di rosso al massimo, verde e blu a zero: rosso puro.
00FF00	Verde	Livello di verde al massimo, rosso e blu a zero: verde puro.
0000FF	Blu	Livello di blu al massimo, rosso e verde a zero: blu puro.
FF9999	Rosa	Livello massimo di rosso, medio di verde e blu.
33CC99	Verde acqua	Livello basso di rosso, alto di verde e medio di blu.
999999	Grigio	Tutti i colori allo stesso livello.

Prof. Montemurro

Tabella dei colori

Colore	Nome simbolico	Red	Green	Blue	Valore esadecimale
	aliceblue	240	248	255	F0F8FF
	antiquewhite	250	235	215	FAEBD7
	aqua	0	255	255	00FFFF
	aquamarine	127	255	212	7FFFDD
	azure	240	255	255	F0FFFF
	beige	245	245	220	F5F5DC
	bisque	255	228	196	FFE4C4
	black	0	0	0	000000
	blanchedalmond	255	235	205	FFEBCD
	blue	0	0	255	0000FF
	blueviolet	138	43	226	8A2BE2
	brown	165	42	42	A52A2A
	burlywood	222	184	135	DEB887
	cadetblue	95	158	160	5F9EA0
	chartreuse	127	255	0	7FFF00
	chocolate	210	105	30	D2691E
	coral	255	127	80	FF7F50
	cornflowerblue	100	149	237	6495ED
	cornsilk	255	248	220	FFF8DC
	crimson	220	20	60	DC143C
	cyan	0	255	255	00FFFF
	darkblue	0	0	139	00008B
	darkcyan	0	139	139	008B8B
	darkgoldenrod	184	134	11	B8860B

Prof. Montemurro

Colore	Nome simbolico	Red	Green	Blue	Valore esadecimale
	darkgray	169	169	169	A9A9A9
	darkgreen	0	100	0	006400
	darkkhaki	189	183	107	BDB76B
	darkmagenta	139	0	139	8B008B
	darkolivegreen	85	107	47	55662F
	darkorange	255	140	0	FF8C00
	darkorchid	153	50	204	9932CC
	darkred	139	0	0	8B0000
	darksalmon	233	150	122	E9967A
	darkseagreen	143	188	143	8FB88F
	darkslateblue	72	61	139	483D8B
	darkslategray	47	79	79	2F4F4F
	darkturquoise	0	206	209	00CED1
	darkviolet	148	0	211	9400D3
	deeppink	255	20	147	FF1493
	deepskyblue	0	191	255	00BFFF
	dimgray	105	105	105	696969
	dodgerblue	30	144	255	1E90FF
	firebrick	178	34	34	B22222
	floralwhite	255	250	240	FFF8A0
	forestgreen	34	139	34	228B22
	fuchsia	255	0	255	FF00FF
	gainsboro	220	220	220	DCDCDC

Prof. Montemurro

Colore	Nome simbolico	Red	Green	Blue	Valore esadecimale
	ghostwhite	248	248	255	F8F8FF
	gold	255	215	0	FFD700
	goldenrod	218	165	32	DAA520
	gray	128	128	128	808080
	green	0	255	0	00FF00
	greenyellow	173	255	47	ADFF2F
	honeydew	240	255	240	F0FFF0
	hotpink	255	105	180	FF69B4
	indianred	205	92	92	CD5C5C
	indigo	75	0	130	4B0082
	ivory	255	255	240	FFFFF0
	khaki	240	230	140	F0E68C
	lavender	230	230	250	E6E6FA
	lavenderblush	255	240	245	FFF0F5
	lawngreen	124	252	0	7CFC00
	lemonchiffon	255	250	205	FFFACD
	lightblue	173	216	230	ADD8E6
	lightcoral	240	128	128	F08080
	lightcyan	224	255	255	E0FFFF
	lightgoldenrodyellow	250	250	210	FAFAD2
	lightgreen	144	238	144	90EE90
	lightgrey	211	211	211	D3D3D3
	lightpink	255	182	193	FFB6C1

Prof. Montemurro

Colore	Nome simbolico	Red	Green	Blue	Valore esadecimale
	lightsalmon	1255	160	122	FFA07A
	lightseagreen	32	178	170	20B2AA
	lightskyblue	135	206	250	87CEFA
	lightslategray	119	136	153	778899
	lightsteelblue	176	196	222	B0C4DE
	lightyellow	255	255	224	FFFFE0
	lime	0	255	0	00FF00
	limegreen	50	205	50	32CD32
	linen	250	240	230	FAF0E6
	magenta	255	0	255	FF00FF
	maroon	128	0	0	800000
	mediumaquamarine	102	205	170	66CDAA
	mediumblue	0	0	205	0000CD
	mediumorchid	186	85	211	BA55D3
	mediumpurple	147	112	219	9370DB
	mediumseagreen	60	179	113	3CB371
	mediumslateblue	123	104	238	7B68EE
	mediumspringgreen	0	250	154	00FA9A
	mediumturquoise	72	209	204	48D1CC
	mediumvioletred	199	21	133	C71585
	midnightblue	25	25	112	191970
	mintcream	245	255	250	F5FFFA
	mistyrose	255	228	225	FFE4E1

Prof. Montemurro

Colore	Nome simbolico	Red	Green	Blue	Valore esadecimale
	moccasin	255	228	181	FFE4B5
	navajowhite	255	222	173	FFDEAD
	navy	0	0	128	000080
	oldlace	253	245	230	FDF5E6
	olive	128	128	0	808000
	olivedrab	107	142	35	6B8E23
	orange	255	165	0	FFA500
	orangered	255	69	0	FF4500
	orchid	218	112	214	DA70D6
	palegoldenrod	238	232	170	EEE8AA
	palegreen	152	251	152	98FB98
	paleturquoise	175	238	238	AEEEEE
	palevioletred	219	112	147	DB7093
	papayawhip	255	239	213	FFEFDS
	peachpuff	255	218	185	FFDAB9
	peru	205	133	63	CD853F
	pink	255	192	203	FFC0CB
	plum	221	160	221	DDA0DD
	powderblue	176	224	230	B0E0E6
	purple	128	0	128	800080
	red	255	0	0	FF0000
	rosybrown	188	143	143	BC8F8F
	royalblue	65	105	225	4169E1

Prof. Montemurro

Colore	Nome simbolico	Red	Green	Blue	Valore esadecimale
	saddlebrown	139	69	19	8B4513
	salmon	250	128	114	FA8072
	sandybrown	244	164	96	F4A460
	seagreen	46	139	87	2E8B57
	seashell	255	245	238	FFF5EE
	sienna	160	82	45	A0522D
	silver	192	192	192	C0C0C0
	skyblue	135	206	235	87CEEB
	slateblue	106	90	205	6A5ACD
	slategray	112	128	144	708090
	snow	255	250	250	FFFFFA
	springgreen	0	255	127	00FF7F
	steelblue	70	130	180	4682B4
	tan	210	180	140	D2B48C
	teal	0	128	128	008080
	thistle	216	191	216	D8BFD8
	tomato	255	99	71	ff6347
	turquoise	64	224	208	40E0D0
	violet	238	130	238	EE82EE
	wheat	245	222	179	F5DEB3
	white	255	255	255	FFFFFF
	whitesmoke	245	245	245	F5F5F5
	yellow	255	255	0	FFFF00
	yellowgreen	154	205	50	9ACD32

Prof. Montemurro

Formattazione di una Pagina Web con HTML

Se le sei cifre esadecimali del colore sono uguali, si può usare una formula abbreviata (**color shorthand**).

Esempi

Forma estesa	Forma abbreviata
#000000	#000
#0000FF	#00F
#33CC66	#3C6

Prof. Montemurro

Formattazione di una Pagina Web con HTML

Colore di sfondo (o **background color**) **di una pagina**: per assegnare un colore allo sfondo di una pagina occorre usare l'attributo **bgcolor** del tag **body**; le sei cifre esadecimali devono essere precedute dal simbolo cancelletto **#**. In alternativa al valore esadecimale, i colori possono essere specificati anche con un **nome simbolico predefinito** (in inglese).

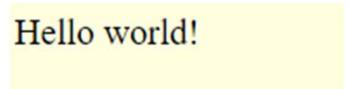
Colore di un testo: per impostare il colore di un testo si usa il tag **font** e l'attributo **color**.

Prof. Montemurro

Formattazione di una Pagina Web con HTML

Esempio: sfondo della pagina giallo chiaro con tag body e attributo bgcolor

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello, World!</title>
  </head>
  <body bgcolor="lightyellow">
    Hello world!
  </body>
</html>
```



Prof. Montemurro

Formattazione di una Pagina Web con HTML

Esempio: colore con tag font e attributo color

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello, World!</title>
  </head>
  <body>
    <font color="dodgerblue">Hello world!</font>
  </body>
</html>
```

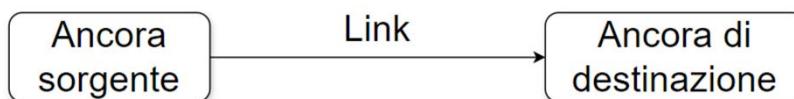


Prof. Montemurro

Link

Link (o **collegamento ipertestuale**, o **iperlink**, o **link web**) in HTML:
connessione tra una risorsa web ed un'altra (es. di risorsa web: pagina web sul mio sito o su un altro sito, email, immagine, video, programma, file ecc.).
 Un link è costituito da tre componenti:

1. un **punto di partenza** (o **ancora sorgente**) che è la parte visibile del link (la parte cliccabile);
2. un **punto di arrivo** (o **ancora di destinazione**, o **ancora target**) che può essere qualunque risorsa web;
3. un **verso** che va dall'ancora sorgente all'ancora di destinazione.



Prof. Montemurro

Link

Il link appare nella pagina come:

1. un'immagine;
2. un'animazione;
3. una parola (o una breve frase) di norma sottolineata e di **colore blu**.

Quando passiamo su un link, il puntatore del mouse si trasforma diventando una piccola mano col dito indice alzato; questo simbolo indica che si può attivare il collegamento cliccando col mouse. Un link già visitato, di solito, assume il colore **rosso porpora**.

Prof. Montemurro

Link

In HTML un link si definisce usando il tag **a** il quale ha la seguente struttura:

```
<a href="nomeRisorsa">parolaCliccabile</a>
```

dove nomeRisorsa è la destinazione del link, e parolaCliccabile è il testo che compare sottolineato e colorato per rappresentare il link; la a del tag sta per anchor, ossia ancora.

Prof. Montemurro

Link

Destinazione del link può essere:

1. un **URL** ad un altro sito web (è sempre un URL assoluto):

```
<a href="https://www.google.it">Vai su Google</a>
```

2. un **URL relativo** per saltare su un'altra pagina del nostro sito il cui file html è nella stessa cartella in cui è presente il file html della pagina che lo richiama:

```
<a href="prova.html">Vai su prova</a>
```

Prof. Montemurro

Link

3. un **URL assoluto** quando si indica il percorso completo per raggiungere il file html della pagina del nostro sito (anche (1) è un URL assoluto):

```
<a href="http://www.scuola1.it/documenti/prova.html">
Vai su prova</a>
```

4. una sezione all'interno della stessa pagina, per poter fare ciò la sezione deve iniziare con un tag HTML identificato con un attributo **id** (univoco nella pagina Web; inoltre **#nome** è chiamato **identificatore di un frammento** e non fa parte dell'URL, ma si aggancia all'URL):

```
<h2 id="sezione2">...</h2>
...
<a href="#sezione2">Vai alla seconda sezione</a>
```

Prof. Montemurro

Link

Per aprire la pagina del link in una nuova finestra del browser si usa l'attributo **target** del tag a con valore **_blank**.

Esempio

```
<a href="https://www.google.it" target="_blank">Vai su Google</a>
```

Prof. Montemurro

Classificazione dei Link

1. **Link interni:** link a pagine i cui file html sono tra quelli del mio sito. Tali link possono utilizzare sia URL assoluti che relativi.
2. **Link esterni:** link a pagine i cui file html sono fuori dal mio sito. Tali link devono necessariamente utilizzare URL assoluti.

Prof. Montemurro

Link

I link interni possono essere utili per organizzare i contenuti di una pagina di grandi dimensioni, inserendo un **indice** iniziale.

```

<h1 id="inizio">Indice dei contenuti</h1>
<p>
  <a href="#sezione1">Introduzione</a><br>
  <a href="#sezione2">Le regole del linguaggio</a><br>
  <a href="#sezione3">Esempi</a><br>
</p>

<h2 id="sezione1">Introduzione</h2>
...
<a href="#inizio">Torna all'indice</a><br>

<h2 id="sezione2">Le regole del linguaggio</h2>
...
<a href="#inizio">Torna all'indice</a><br>

<h2 id="sezione3">Esempi</h2>
...
  
```

Le sezioni

L'id `inizio` per dare la possibilità, alla fine di ogni sezione, di tornare all'indice

Ogni volta che c'è href c'è un salto.

Prof. Montemurro

Link alla Sezione di un'Altra Pagina Interna

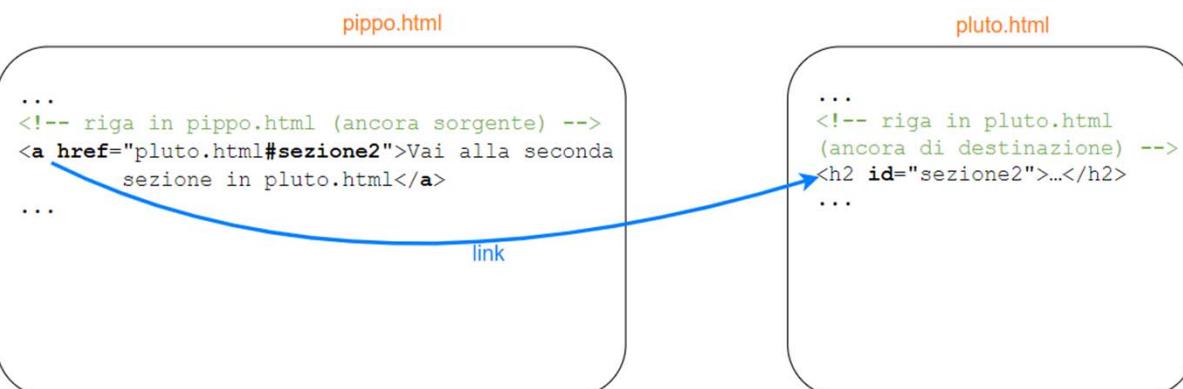
Date due pagine **pippo.html** e **pluto.html**, entrambe appartenenti allo stesso nostro sito (www.sito.it), si vuole saltare alla sezione2 della pagina interna **pluto.html** partendo dalla pagina **pippo.html** (posso usare un URL relativo o assoluto).

```
<!-- riga in pluto.html (ancora di destinazione) -->
<h2 id="sezione2">...</h2>

<!-- riga in pippo.html (ancora sorgente) -->
<a href="pluto.html#sezione2">Vai alla seconda
sezione in pluto.html</a>
```

Prof. Montemurro

Link alla Sezione di un'Altra Pagina Interna



Prof. Montemurro

Link alla Sezione di una Pagina Esterna

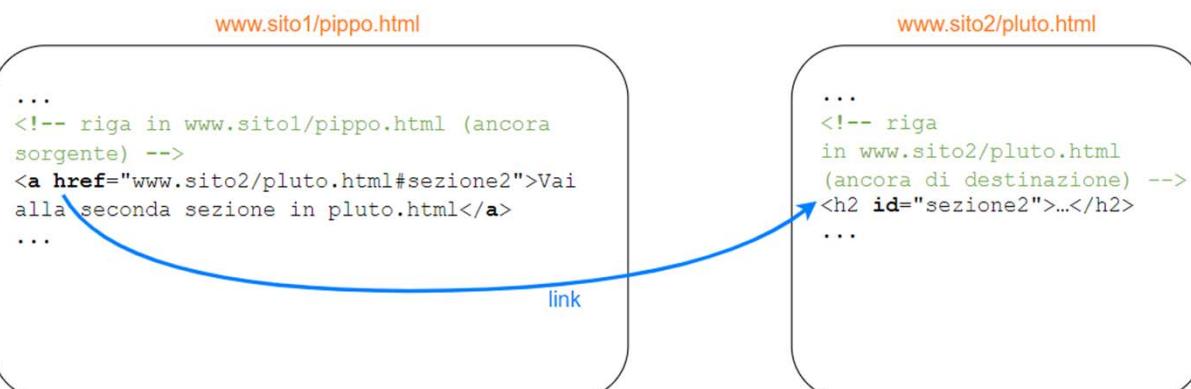
Date due pagine [www.sito1/pippo.html](#) e [www.sito2/pluto.html](#), appartenenti a siti diversi (quello nostro in cui scriviamo il tag , e quello esterno in cui c'è già l'`id` alla sezione desiderata), si vuole saltare alla sezione2 della pagina esterna [pluto.html](#) partendo dalla pagina [pippo.html](#) (devo usare URL assoluto).

```
<!-- riga in www.sito2/pluto.html (ancora di destinazione) -->
<h2 id="sezione2">...</h2>

<!-- riga in www.sito1/pippo.html (ancora sorgente) -->
<a href="www.sito2/pluto.html#sezione2">Vai alla seconda sezione in pluto.html</a>
```

Prof. Montemurro

Link alla Sezione di una Pagina Esterna



Prof. Montemurro

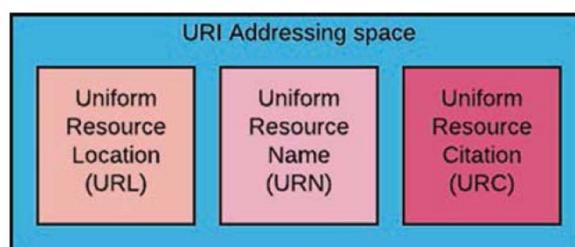
$URI = URL \cup URN \cup URC$

1. **Uniform Resource Identifier (URI)**: è l'insieme di tutti i nomi/indirizzi, costituito da stringhe brevi di caratteri, usato per identificare le risorse web.
 - i. **Risorsa fisica** (es. pagina web): risorsa identificata mediante il suo indirizzo.
 - ii. **Risorsa astratta** (es. ISBN di un libro): risorsa identificata mediante il suo nome univoco.
2. **Uniform Resource Location (URL)**: è un URI che identifica una risorsa mediante il suo indirizzo per cui, se la risorsa si muove, il suo URL cambia.
3. **Uniform Resource Name (URN)**: è un URI che identifica una risorsa mediante il suo nome univoco.
4. **Uniform Resource Citation (*)**(poi diventato **Uniform Resource Characteristics (URC)**): è un URI costituito da un insieme di attributi/valori (metadati) che descrive una risorsa (autore, editore, diritti d'autore ecc.).

(*) Oggi si usano altri standard come Dublin Core e RDF (Resource Description Framework).

Prof. Montemurro

$URI = URL \cup URN \cup URC$



Analogia considerando la risorsa "persona"

1. URI: insieme di codice fiscale, indirizzo di casa, caratteristiche fisiche
2. URL: indirizzo di casa (oggi meglio coordinate GPS dello smartphone)
3. URN: codice fiscale oppure numero della carta d'identità o della patente
4. URC: caratteristiche fisiche

Prof. Montemurro

IRI

Oggi, oltre all'URI, si usa anche l'**IRI** (**Internationalized Resource Identifier**) il quale, a differenza dell'URI, ammette anche l'uso di caratteri appartenenti ad alfabeti non latini.

Attualmente la maggior parte dei meccanismi di recupero delle risorse sono definiti solo per gli URI, ecco perché si rende necessario avere una relazione tra gli IRI e gli URI.

Prof. Montemurro

Link ad un Indirizzo di Posta Elettronica

In genere su una pagina web (specialmente quella dei contatti) si trova un collegamento ipertestuale all'indirizzo di posta elettronica dell'autore del sito. Per inserire tale collegamento si usa lo **schema mailto** dell'URL:

```
<a href="mailto:rossi@societax.it">Scrivete a Rossi  
Mario</a>
```

Cliccando sul link viene aperto un modulo vuoto di messaggio di posta elettronica (in genere quella predefinita come Google Gmail o Microsoft Outlook) che sarà inviato all'indirizzo email indicato in mailto.

Prof. Montemurro

Link ad un Indirizzo di Posta Elettronica

Collegamento ipertestuale all'indirizzo email di più destinatari:

```
<a href="mailto:utentel@esempio.it,utente2@esempio.it">
Email di gruppo</a>
```

Prof. Montemurro

Link ad un Indirizzo di Posta Elettronica

Nello **schema mailto** dell'URL si possono inserire anche altri parametri (%20 significa spazio vuoto; ? per eseguire una query, cioè invia informazioni aggiuntive all'applicazione/servizio web; & sta per and):

1. l'oggetto dell'email tramite il parametro **subject**;
2. il corpo dell'email tramite il parametro **body**;
3. uno o più destinatari indiretti tramite il parametro **cc** (Copia per Conoscenza CC) o **bcc** (Copia per Conoscenza Nascosta NCC).

```
<a href=
mailto:rossi@societaX.it?subject=Oggetto%20messaggio
&body=How%20are%20you?&cc=pippo@esempio.it&bcc=pluto
@esempio.it">Scrivete a Rossi Mario, CC a Pippo, CCN
a Pluto</a>
```

Prof. Montemurro

Schemi URL, Schemi URI

Schema URL/URI: è un componente logico della struttura dell'URL/URI che definisce il protocollo (insieme di regole) o il meccanismo usato per accedere ad una risorsa.

Sintassi (o struttura) generale di un URI/URL

URI = schema ":" parte-specifica-schema

Esempio

Schema **whatsapp** permette di iniziare una chat col numero di cellulare specificato:

`whatsapp://send?phone=1234567890`

Prof. Montemurro

Schemi URL, Schemi URI

Sintassi (o struttura) generale di un URL



1. Schema (obbligatorio)
 - i. schemi comuni: http, https, mailto, ftp, tel, file, data, sms, sip, ws, wss, irc, rtsp, geo, ssh, sftp, magnet, blob
 - ii. schemi specializzati: bitcoin, idap, gopher, chrome, about, intent, telnet, market, javascript, vnc, skype, zoomtg, whatsapp

Prof. Montemurro

Schemi URL, Schemi URI

Sintassi (o struttura) generale di un URL



2. Autorità (opzionale)

authority = [userinfo@] host [: port]

dove:

- i. userinfo riguarda informazioni di autenticazione (es. username:password@)
- ii. host specifica il nome del dominio, l'indirizzo IP, l'indirizzo IPv6
- iii. port specifica il numero di **porta** la quale indica indirizzo per identificare univocamente un'applicazione sul computer

Prof. Montemurro

Schemi URL, Schemi URI

Sintassi (o struttura) generale di un URL



3. Percorso (opzionale) il quale ha una struttura gerarchica o meno e serve per identificare una risorsa
4. Parametri di query (opzionali) i quali servono per inviare informazioni aggiuntive all'applicazione/servizio web; notare **?** all'inizio della query
5. Identificatore di un frammento (opzionale) **#nome** il quale si riferisce ad una risorsa secondaria all'interno di una risorsa primaria (es. una sezione all'interno di una pagina web)

Prof. Montemurro

Liste Puntate e Liste Numerate

Sintassi **lista puntata**:

- si apre la lista col tag di apertura **** (**unordered list**);
- si inserisce la coppia di tag **...** (**list item**) per ogni riga della lista;
- si chiude la lista col tag ****.

Sintassi **lista numerata**: come lista puntata, solo che i tag **...** vanno sostituiti con i tag **...** (**ordered list**).

Prof. Montemurro

Esempio Lista Puntata

```
<!DOCTYPE html>
<html>
  <head>
    <title>Lista puntata</title>
  <body>
    <ul>
      <li>Roma</li>
      <li>Milano</li>
      <li>Napoli</li>
      <li>Firenze</li>
      <li>Bologna</li>
      <li>Venezia</li>
    </ul>
  </body>
</html>
```

- Roma
- Milano
- Napoli
- Firenze
- Bologna
- Venezia

Prof. Montemurro

Esempio Lista Numerata

```
<!DOCTYPE html>
<html>
  <head>
    <title>Lista puntata</title>
  <body>
    <ol>
      <li>Roma</li>
      <li>Milano</li>
      <li>Napoli</li>
      <li>Firenze</li>
      <li>Bologna</li>
      <li>Venezia</li>
    </ol>
  </body>
</html>
```

1. Roma
2. Milano
3. Napoli
4. Firenze
5. Bologna
6. Venezia

Prof. Montemurro

Tabelle

Sintassi **tabella**:

- si apre la tabella col tag di apertura **<table>**;
- si inserisce la coppia di tag **<tr>...</tr>** (**table row**) per ogni riga della tabella;
- all'interno della riga, ogni cella è racchiusa tra la coppia di tag **<td>...</td>** (**table data**);
- si chiude la tabella col tag di chiusura **</table>**.

Le celle della riga di intestazione delle colonne, con i caratteri in grassetto, sono identificate dai tag **<th>...</th>** (**table header**); questi vanno messi tra i tag **<tr>...</tr>** della prima riga.

Prof. Montemurro

Tabelle

In una pagina Web, le tabelle vengono usate in due modi:

1. per organizzare i dati su righe e colonne, secondo una griglia che facilita la visualizzazione delle informazioni;
2. per impaginare testo e immagini in modo ordinato; in questo caso i bordi delle celle devono essere assenti e la tabella viene usata solo come traccia per disporre gli elementi.

Prof. Montemurro

Tabelle

Attributi comuni del tag `table`:

1. **`border`**= "numeroDiPixel" per aggiungere il bordo alla tabella;
 - `numeroDiPixel=0` per tabella senza bordo;
 - `numeroDiPixel>0` per tabella con bordo; maggiore è `numeroDiPixel`, maggiore è lo spessore del bordo;
2. **`width`**= "valoreInPercentuale" (es. 50%)
`width= "numeroDiPixel"` (es. 150)
 per stabilire la percentuale di occupazione della pagina da parte della tabella in termini di larghezza della tabella stessa;
3. **`cellspacing`**= "numeroDiPixel" per modificare lo spazio tra i bordi delle celle, cioè (i) tra i bordi di due celle adiacenti, e (ii) tra il bordo delle celle periferiche ed il bordo della tabella;
4. **`cellpadding`**= "numeroDiPixel" per modificare lo spazio tra il contenuto delle celle ed i bordi interni.

Prof. Montemurro

Tabelle

Attributo comune del tag th o td:

width=valoreInPercentuale (es. 50%)

oppure

width=numeroDiPixel (es. 150)

per fissare la larghezza di ciascuna colonna espressa in percentuale oppure in numero di pixel.

Prof. Montemurro

Esempio Tabella con Cellspacing e Cellpadding

```
<table border="1" cellspacing="0" cellpadding="0">
  <tr>
    <td>Primavera</td>
    <td>Estate</td>
  </tr>
  <tr>
    <td>Autunno</td>
    <td>Inverno</td>
  </tr>
</table>
```

Prof. Montemurro

Table 1:

Primavera	Estate
Autunno	Inverno

Table 2:

Primavera	Estate
Autunno	Inverno

Table 3:

Primavera	Estate
Autunno	Inverno

Table 4:

Primavera	Estate
Autunno	Inverno

Prof. Montemurro

Tabelle: Unione delle Celle

I tag `<td>` e `<th>` hanno due attributi, **colspan (column span)** e **rowspan (row span)**, che permettono di unire più celle tra loro.

1. **colspan=numeroColonne** significa che la **cella** definita col tag `<td>` o `<th>` occuperà un numero di colonne pari a numeroColonne; dunque tale **cella** deriva dall'unione di un numero di **celle orizzontali** pari a numeroColonne;
2. **rowspan=numeroRighe** significa che la **cella** definita col tag `<td>` o `<th>` occuperà un numero di righe pari a numeroRighe; dunque tale **cella** deriva dall'unione di un numero di **celle verticali** pari a numeroRighe.

Prof. Montemurro

Tabelle: Unione delle Celle

Esempio 1: tabella 2x3 con prima cella della prima riga che occupa 3 colonne.

```
<table width="200" border="1">
  <tr>
    <th colspan="3">Primo trimestre</th>
  </tr>
  <tr>
    <td>Gennaio</td>
    <td>Febbraio</td>
    <td>Marzo</td>
  </tr>
</table>
```

Primo trimestre		
Gennaio	Febbraio	Marzo

Prof. Montemurro

Tabelle: Unione delle Celle

Esempio 2: tabella 6x3 con prima cella della prima colonna che occupa 4 righe, e quinta cella della prima colonna che occupa 2 righe.

	Zona	Agente
Italia	Nord	Rossi
	Centro	Bianchi
	Sud e Isole	Verdi
	Europa	Smith
Estero	Asia	Ching Sai

Prof. Montemurro

Tabelle: Unione delle Celle

```
<table width="200" border="1">
  <tr>
    <th rowspan="4">Italia</th>
    <th>Zona</th>
    <th>Agente</th>
  </tr>
  <tr>
    <td>Nord</td>
    <td>Rossi</td>
  </tr>
  <tr>
    <td>Centro</td>
    <td>Bianchi</td>
  </tr>
  <tr>
    <td>Sud e Isole</td>
    <td>Verdi</td>
  </tr>
```

```
<tr>
  <th rowspan="2">Ester</th>
  <td>Europa</td>
  <td>Smith</td>
</tr>
<tr>
  <td>Asia</td>
  <td>Ching Sai</td>
</tr>
</table>
```

	Zona	Agente
Italia	Nord	Rossi
	Centro	Bianchi
	Sud e Isole	Verdi
Ester	Europa	Smith
	Asia	Ching Sai

Prof. Montemurro

1 Elementi Multimediali: Immagini

Per inserire un'immagine si usa il tag **** (image) con l'attributo **src** (source), che serve a specificare il percorso dell'immagine, ed eventuali altri attributi:

```

```

tale tag è un tag vuoto in quanto non esiste il tag di chiusura per cui può essere scritto anche nella forma con la barra finale:

```

```

Prof. Montemurro

1 Elementi Multimediali: Immagini

Attributi del tag :

1. **src**= "PercorsoFileImmagine" per specificare il percorso (URL) del file immagine;
2. **border**= "numeroDiPixel" per specificare lo spessore del bordo dell'immagine;
3. Per specificare la larghezza dell'immagine:
width= "numeroDiPixel"
width= "valoreInPercentuale" rispetto alla larghezza complessiva della pagina
4. per specificare l'altezza dell'immagine:
height= "numeroDiPixel"
height= "valoreInPercentuale" rispetto all'altezza complessiva della pagina

Prof. Montemurro

1 Elementi Multimediali: Immagini

5. **align**= "tipoAllineamento" per allineare l'immagine al testo o ad un'altra immagine; in tipoAllineamento ci va uno dei seguenti valori testuali:
 i. **middle** per avere un allineamento centrato;
 ii. **right** per avere un allineamento a destra;
 iii. **left** per avere un allineamento a sinistra;
 iv. **bottom** per avere un allineamento in basso;
 v. **top** per avere un allineamento in alto;
6. **alt**= "fraseAlternativa" la quale può assolvere una delle seguenti due funzioni:
 - i. mostrare la frase scritta quando il browser non riesce a visualizzare correttamente l'immagine;
 - ii. far apparire un commento quando il mouse passa sopra l'immagine.

Prof. Montemurro

1 Elementi Multimediali: Immagini

Formati delle immagini più comuni supportati dai browser:

1. **GIF (Graphics Interchange Format)**: animazioni, sfondo trasparente, 256 colori
2. **JPEG (o JPG) (Joint Photographic Expert Group)**: 16 milioni di colori, no animazioni, no sfondo trasparente
3. **PNG (Portable Network Graphics)**: ibrido tra GIF e JPEG, alta qualità, sfondo trasparente

Prof. Montemurro

2 Elementi Multimediali: Video

Per inserire un **video** si utilizza:

1. la coppia di tag **<video>...</video>** in cui, tramite l'attributo **poster** si specifica il percorso dell'immagine (URL) che viene visualizzata prima dell'avvio del video; l'attributo **controls** mostra i **controlli di riproduzione** (play, pausa, regolazione del volume, barra di avanzamento);
2. il tag **source**, che è un tag vuoto, in cui, tramite l'attributo **src**, si specifica il percorso del video (URL video), e l'attributo **type** specifica il formato del video (es. `video/mp4`, `video/ogg`, `video/webm`).

```
<video poster="percorsoFileCopertina" controls>
  <source src="percorsoFileVideo"
    type="video/formato">
  messaggio errore
</video>
```



Prof. Montemurro

2 Elementi Multimediali: Video

```
<video poster="percorsoFileCopertina" controls>
  <source src="percorsoFileVideo"
    type="video/formato">
  messaggio errore
</video>
```

dove messaggio errore è un testo che viene mostrato se il browser non è compatibile col tag video.

Prof. Montemurro

2 Elementi Multimediali: Video

Attributi comuni del tag **video**:

1. **controls** per mostrare i controlli di riproduzione;
2. **autoplay** per avviare il file video automaticamente;
3. **loop** per ripetere il video più volte automaticamente;
4. **preload** per caricare il video prima dell'avvio.

I suddetti quattro attributi sono di tipo booleano per cui i seguenti codici sono equivalenti:

```
<video controls>
<video controls="controls">
<video controls=" ">
```

Prof. Montemurro

2 Elementi Multimediali: Video

Caso generale: più tag source all'interno di un unico tag video

```
<video poster="percorsoFileCopertina" controls>
  <source src="percorsoFileVideo1"
    type="video/formato1">
  <source src="percorsoFileVideo2"
    type="video/formato2">
  messaggio errore
</video>
```

Il browser legge automaticamente il primo source che supporta tra l'elenco dei source che abbiamo scritto.

Prof. Montemurro

2 Elementi Multimediali: Video

Vantaggi del caso generale

1. **Flessibilità del formato:** ci sono dei browser che non supportano tutti i formati video. Per garantire che un video venga riprodotto da qualunque browser, occorre inserire più volte lo stesso video, ma ogni volta con un formato diverso (es. mp4, ogg ecc.).
2. **Flessibilità della risoluzione:** a seconda della velocità di connessione e del dispositivo usato, i siti web possono offrire risoluzioni video diverse (se la connessione è lenta, converrà usare una risoluzione più bassa altrimenti il video va a scatti o non parte).
3. **Flessibilità del codec (coder-encoder, o codificatore-decodificatore):** alcuni dispositivi funzionano meglio con codec specifici, quindi fornire più opzioni di codifica (ad ognuna è associata un source) può migliorare le prestazioni.

Prof. Montemurro

2 Elementi Multimediali: Video

Codec: software (*) in grado di codificare o decodificare segnali audio e video attraverso un processo di digitalizzazione, compressione e decompressione, al fine di:

1. ridurre lo spazio di memorizzazione occupato dai dati che si vogliono salvare su un supporto di memorizzazione;
2. ridurre l'occupazione della banda (**) occupata dai dati che si vogliono trasferire tra reti di calcolatori.

(*) Nel nostro caso il codec è un software, ma, in generale, può essere anche un dispositivo hardware.

(**) **Banda:** tasso di informazione trasmessa da una connessione di rete o da un'interfaccia (es. modem 56k ha una banda di 56 kilobit al secondo kbps).

Prof. Montemurro

2 Elementi Multimediali: Video

Codec	Best For	Pros	Cons	Browser Support
H.264	General web usage	Widely supported, good quality, decent size	Larger file size than newer codecs	<input checked="" type="checkbox"/> All browsers
H.265 (HEVC)	High-quality streaming	Better compression than H.264	Not supported in all browsers	<input type="checkbox"/> Limited (Safari, Edge)
VP9	Web streaming (YouTube, etc.)	Open-source, high quality, better compression	Needs higher processing power	<input checked="" type="checkbox"/> Chrome, Firefox, Edge
AV1	Future-proof high-quality streaming	Best compression, royalty-free	Not widely supported yet	<input checked="" type="checkbox"/> Chrome, Firefox (Partial)
Theora	Older open-source alternative	Free to use	Poorer compression than H.264	<input type="checkbox"/> Limited (Firefox)

Prof. Montemurro

2 Elementi Multimediali: Video

Per inserire un video di YouTube o un generico video in una pagina web si può usare la coppia di tag <**iframe**>...<**iframe**/>:

```
<iframe src="indirizzo" height="altezzaInPixel"
       width="larghezzaInPixel">
messaggio errore: tag iframe non supportato
</iframe>
```

In questo modo il video non viene realmente inserito nella pagina web, ma si crea nella pagina una finestra che contiene, al suo interno, un video o una qualsiasi altra risorsa Web.

Nota bene: YouTube non consente di inserire i video coi tag **video** e **source**.

Prof. Montemurro

2 Elementi Multimediali: Video

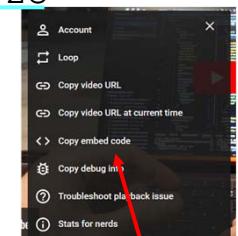
Esempio

 youtube.com/watch?v=CNsvts6pVzo&ab_channel=EdmundYong

IDVideo (tra = e &)

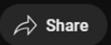
```
<iframe
src="https://www.youtube.com/embed/CNsvts6pVzo"
width="854" height="480" frameborder="0"
allowfullscreen></iframe>
```

1. Si crea una finestra di dimensione 854 x 480, senza bordi (**frameborder="0"**).
2. L'attributo **allowfullscreen** permette all'utente di visualizzare il video a schermo intero.
3. È possibile ottenere il tag completo cliccando con il tasto destro sul video di YouTube e selezionando l'opzione **Copia codice per incorporamento (Copy embed code)**.



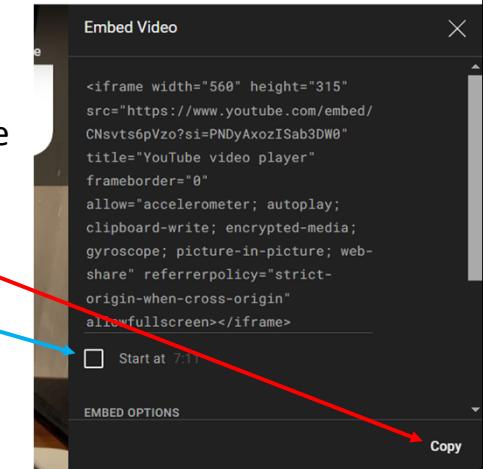
Prof. Montemurro

2 Elementi Multimediali: Video

...continuazione esempio: un modo più semplice per inserire un video
consiste nel cliccare su  e poi su 

infine su **Copy**.

Start at permette di far partire il video dall'istante
desiderato.



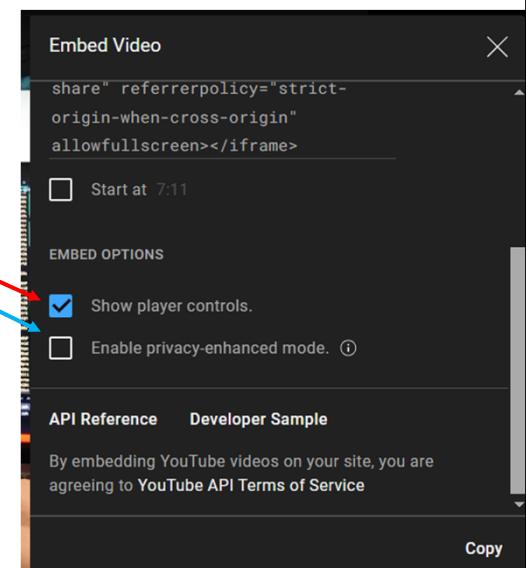
2 Elementi Multimediali: Video

...continuazione esempio

Show player controls: mostra i controlli di
riproduzione.

Enable privacy-enhanced mode: impedisce a
YouTube di personalizzare:

1. l'esperienza di navigazione dell'utente
su YouTube;
2. gli eventuali annunci che compaiono nel
video.



3 Elementi Multimediali: Audio

Per inserire un audio si utilizza:

1. la coppia di tag `<audio>...</audio>`; l'attributo **controls** mostra i **controlli di riproduzione** (play, pausa, regolazione del volume, barra di avanzamento);
2. il tag **source**, che è un tag vuoto, in cui, tramite l'attributo **src**, si specifica il percorso del file audio (URL audio), e l'attributo **type** specifica il formato del file audio (es. video/mpeg, video/ogg ecc.).

```
<audio controls>
  <source src="percorsoFileAudio"
    type="audio/formato">
  messaggio errore
</audio>
```

Prof. Montemurro

3 Elementi Multimediali: Audio

```
<audio controls>
  <source src="percorsoFileAudio"
    type="audio/formato">
  messaggio errore
</audio>
```

dove `messaggio errore` è un testo che viene mostrato se il browser non è compatibile col tag `audio`.

Prof. Montemurro

3 Elementi Multimediali: Audio

Attributi comuni del tag **audio**:

1. **controls** per mostrare i controlli di riproduzione;
2. **autoplay** per avviare il file audio automaticamente; potrebbe essere utile per inserire **musica di sottofondo** (**controls** va tolto);
3. **loop** per ripetere l'audio più volte automaticamente;
4. **preload** per caricare l'audio prima dell'avvio.

I suddetti quattro attributi sono di tipo booleano per cui i seguenti codici sono equivalenti:

```
<audio controls>
<audio controls="controls">
<audio controls=" ">
```

Prof. Montemurro

3 Elementi Multimediali: Audio

Per inserire un file audio o una playlist in una pagina web si può usare la coppia di tag **<iframe>...</iframe>**.

Il seguente codice permette di inserire un file audio o una playlist direttamente da Spotify:

```
<iframe
  src="https://open.spotify.com/embed?uri=spotify:us
  er:spotify:
  playlist:IDaudio" frameborder="0" allowfullscreen>
</iframe>
```

Anche in questo caso, per ottenere il tag, si può cliccare con il tasto destro sul brano e selezionare l'opzione **Copia link brano**.

Prof. Montemurro

Modi di Inserire Spazi in HTML

1. per inserire 1 spazio (attenzione a scrivere il `i`)
 per inserire 2 spazi
 per inserire $2 + 1 = 3$ spazi
 per inserire 4 spazi
 ...
2. Tag `<pre></pre>` il quale inserisce almeno 2 righe vuote più eventuali spazi e/o altre righe vuote.
3. `
` per inserire 1 riga vuota
`

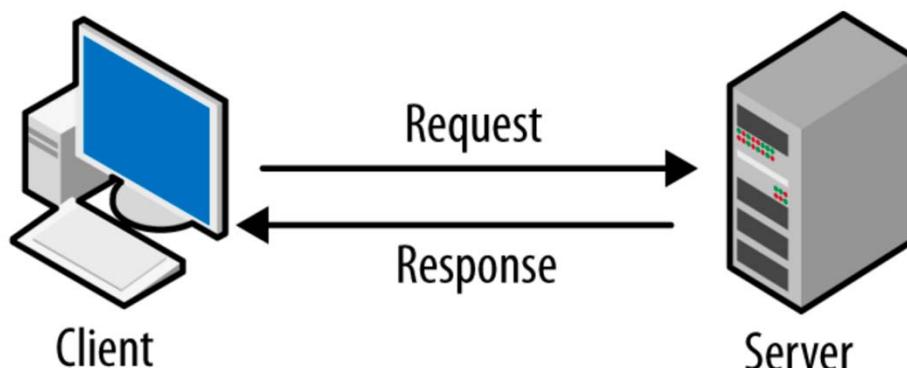
` per inserire 2 righe vuote
 ...
4. Con la proprietà CSS `style="text-indent:numeroPixel"` dove numeroPixel (es. `100px`) è lo spazio espresso in termini di numero di pixel...lo vedremo più avanti.

Prof. Montemurro

Architettura Client-Server

Client: applicazione che richiede un servizio (es. Gmail sul browser del nostro PC).

Server: applicazione che fornisce un servizio (es. server Google fornisce Gmail).



Prof. Montemurro

Form HTML

Form (o modulo HTML): elemento HTML che permette l'invio di informazioni dal client al server; tali informazioni (dati inseriti dall'utente e/o scelte fatte dall'utente) sono inserite dall'utente nel form tramite il browser. Un form è rappresentato dai tag `<form>...</form>`. I dati sono inseriti in appositi spazi del form chiamati **campi** (del form).

I tag `<form>...</form>` vanno messi nei tag `<body>...</body>`.

Prof. Montemurro

Form HTML Dati personali

Esempio

The diagram illustrates an HTML form for personal data with the following components:

- Casella di testo (Text Input):** Three stacked input fields for Name, Surname, and Password.
- Pulsante di opzione radio (Radio Button):** A group of radio buttons for selecting a credit card type, with "Visa" selected.
- Area di testo (Text Area):** A large input field for writing a question.
- Menù a tendina (Select):** A dropdown menu for choosing a region in Italy, currently set to "nord".
- Casella di controllo (Checkbox):** A checkbox for indicating known languages, with "italiano" checked.
- Pulsante di invio dei dati (Submit Button):** A button labeled "Invia".
- Pulsante di cancellazione dei dati (Delete Button):** A button labeled "Cancella i dati".

Prof. Montemurro

Form HTML

Tag <input>: usato per aggiungere i componenti del form usati per inserire un valore o per effettuare una scelta.

Attributi comuni del tag <input>

- **type** per specificare il tipo di input (ess. text, submit, reset, password, hidden, button, radio, checkbox)
- **name** per identificare un oggetto all'interno del form
- **value** per specificare il valore di un elemento input
- **size** per indicare le dimensioni in termini di "numero di caratteri visibili" di una casella di testo

Prof. Montemurro

Form HTML

Esempio dati inviati qui sul server quando si clicca sul pulsante *Invia* (di tipo submit)

```
<form action="URL" method="post">
    Nome <input type="text" name="nomePersona"
           size="25"><br>
    Cognome <input type="text" name="cognomePersona"
              size="30"><br>
    <input type="submit" name="invia" value="Invia">
    <input type="reset" name="annulla" value="Cancella
          i dati" size="14">
</form>
```

Nome	<input type="text"/>
Cognome	<input type="text"/>
	<input type="button" value="Invia"/> <input type="button" value="Cancella i dati"/>

Prof. Montemurro

Form HTML

action: attributo del tag `form` che indica dove si trova il programma PHP sul server che riceverà ed elaborerà i campi del form.

method: attributo del tag `form` che specifica (al browser) la modalità con cui inviare i campi del form al server. Esistono due metodi, get e post, le cui differenze sono:

GET	POST
Dati inviati <u>visibili</u> nell'URL	Dati inviati <u>non</u> visibili nell'URL
Pagina generata <u>salvabile</u> nei preferiti	Pagina generata <u>non</u> salvabile nei preferiti
Massima lunghezza URL: 3000 caratteri	URL di <u>qualsiasi</u> lunghezza

Nota bene: get non adatto per l'invio di dati sensibili (es. password, numero di carta di credito).

Prof. Montemurro

Tipi del Tag `input` del `form`

- **text** per le caselle di testo; l'attributo **value** per precompilare il campo del form
- **submit** per il pulsante testuale di invio dei dati
- **reset** per il pulsante di cancellazione dei dati
- **image** per inserire un pulsante grafico di invio dei dati (è uguale al tipo submit, ma qui si usa un'immagine invece che il testo)
- **password** per le caselle password; attributo **size** specifica il numero di caratteri (nascosti da puntini) che entrano nella casella visibile
- **hidden** per i campi non visibili e quindi non modificabili dall'utente
- **button** per inserire un generico pulsante di comando che serve per avviare l'esecuzione di istruzioni di elaborazione o di controllo

Prof. Montemurro

Tipi del Tag input del form

- **radio** per inserire un pulsante di opzione che consente di effettuare una sola scelta all'interno di un elenco di voci indicate con piccoli cerchi. L'eventuale elemento preselezionato è indicato con l'attributo **checked**. Tutti gli oggetti dell'elenco devono avere lo stesso valore per l'attributo **name**
- **checkbox** per inserire una casella di controllo utilizzata per selezionare anche più di una voce contemporaneamente in un elenco di voci indicate con piccoli quadrati. L'eventuale elemento preselezionato è indicato con l'attributo **checked**
- **email** per inserire una o più email
- **date** per inserire una data
- **datetime-local** per inserire data e ora

Prof. Montemurro

Tipi del Tag input del form

- **time** per inserire un'ora (ora e minuti, e opzionalmente secondi)
- **url** per inserire un URL
- **number** per inserire un valore numerico; gli attributi **min** e **max** servono per specificare l'intervallo di valori validi
- **tel** per inserire un numero di telefono
- **range** per visualizzare un **selettore** che permette di scegliere un valore compreso in un intervallo
- **color** per selezionare un colore
- **file** per caricare uno più file da inviare al server

Prof. Montemurro

Attributo checked del Tag input di Tipo radio e checkbox

checked: attributo booleano (cioè o vale true o false) del tag input di tipo radio e checkbox usato per preselezionare un elemento.

Attenzione: l'uso corretto dell'attributo checked prevede di non assegnargli alcun valore, nonostante sia un attributo booleano:

`checked`

Dunque è sbagliato assegnargli il valore true o false o un qualsiasi altro valore in quanto le seguenti scritture equivalgono a quella senza valore (cioè a `checked`):

`checked="true"`
`checked="false"`
`checked="pippo"`

Prof. Montemurro

Form HTML: Attributo value del Tag input

value: attributo per specificare il valore di un elemento **input**. In particolare:

1. se il tipo del tag input è "button", "reset", "submit", allora value indica la scritta che compare sul pulsante;
2. se il tipo del tag input è "text", "password", "hidden", allora value serve per specificare il valore predefinito del campo di input;
3. se il tipo del tag input è "checkbox", "radio", allora value definisce i dati che saranno inviati al server quando l'utente cliccherà sul pulsante di invio dei dati.

Prof. Montemurro

Form HTML: Tipo text del Tag input

text: tipo del tag `input` per inserire le caselle di testo. L'attributo `value` nel tag `input` di tipo `text` permette di precompilare il campo del form specificando un valore predefinito.

Esempio

```
Nome1 <input type="text" name="nome1Persona"
           size="25"><br>
Nome2 <input type="text" name="nome2Persona"
           size="25" value="Pippo"><br>
```

Nome1	<input type="text"/>
Nome2	<input type="text" value="Pippo"/>

Prof. Montemurro

Form HTML: Attributo placeholder del Tag input

placeholder: attributo per specificare un breve suggerimento che descrive il valore/formato atteso in un campo `input` (es. si può specificare un valore campione o una breve descrizione del formato del dato da inserire nel campo).

Esempio

```
<input type="text" name="nomePersona"
       size="25" placeholder="Inserisci nome persona"><br>
<input type="text" name="nomePianta"
       size="25" placeholder="Inserisci nome pianta"><br>
```

Inserisci nome persona
Inserisci nome pianta

Nota bene: quando l'utente clicca sul campo per inserire il testo, non deve cancellare il testo in grigio, quello specificato con `placeholder`, in quanto tale testo scompare quando l'utente inizia a scrivere nella casella di testo.

Prof. Montemurro

Form HTML: Tag label

label: tag per descrivere gli elementi di un form in modo efficiente; tale tag è importante per favorire l'**accessibilità** dei siti Web anche per **persone con disabilità**:

1. i software che leggono il contenuto dello schermo (screen reader) per gli utenti con **difficoltà visive** possono leggere ad alta voce la **label** quando l'utente passa con il mouse sopra l'elemento;
2. gli utenti con **difficoltà nel muovere il mouse** possono cliccare su un'area più grande, che comprende anche la **label**, per selezionare un elemento del form, ad essa associato, di dimensioni ridotte.

Tag associabili col tag label: <button>, <input> (tranne tipo hidden), <select>, <textarea>, <date> ed altri non appartenenti al form (es. <meter>, <output>, <progress>).

Prof. Montemurro

Form HTML: Tag label del form

Esempio attributo che specifica a quale elemento del form è associata la label

```
<label for="nomePersona">Nome</label>
<input type="text" name="nomePersona" id="nomePersona"
size="25">
```

oppure tag input dentro tag label (attributo for e id non servono più):

```
<label>
  Nome <input type="text" name="nomePersona" size="25">
</label>
```

Nome

Nota bene: grazie al tag label, l'utente può scrivere nella casella di testo o cliccando sulla casella di testo o anche cliccando su Nome.

Prof. Montemurro

Form HTML: Tipo submit del Tag input

submit: tipo del tag `input` per inserire un pulsante testuale di invio dei dati. Se il testo è lungo e se si vuole il testo su un'unica riga, usare l'attributo `size` per stabilire quanti caratteri possono entrare su una singola riga.

Esempio

```
<input type="submit" name="invia" value="Invia"> 
```

Prof. Montemurro

Form HTML: Tipo reset del Tag input

reset: tipo del tag `input` per inserire il pulsante di cancellazione dei dati. Se il testo è lungo e se si vuole il testo su un'unica riga, usare l'attributo `size` per stabilire quanti caratteri possono entrare su una singola riga.

Esempio

```
<input type="reset" name="annulla" value="Cancella  
i dati" size="14"> 
```

Prof. Montemurro

Form HTML: Tipo image del Tag input

image: tipo del tag `input` per inserire un pulsante grafico di invio dei dati.

Esempio

```
<input type="image" src="percorsoImmagine"
       alt="Invia" width="100" height="100">
```



alt="fraseAlternativa" serve o per:

- i. mostrare la frase scritta quando il browser non riesce a visualizzare correttamente l'immagine;
- ii. far apparire un commento quando il mouse passa sopra l'immagine.

Prof. Montemurro

Form HTML: Tipo password del Tag input

Esempio

```
<input type="password" name="pwd" size="20"
       maxlength="12", placeholder="Password">
```


<!--

maxlength indica il massimo numero di caratteri di cui può essere composta la password

size indica il numero di caratteri (nascosti da puntini) che potenzialmente entrano nella casella visibile

-->

Prof. Montemurro

Form HTML: Tipo button del Tag input

button: tipo del tag `input` per inserire un generico pulsante di comando che serve per avviare l'esecuzione di istruzioni di elaborazione o di controllo.

Esempio

```
<input type="button" value="Salva" ...>
<!--
... indicano che ci sono altri attributi che servono
per specificare quali azioni eseguire quando si
clicca il pulsante di comando
-->
```

Salva

Prof. Montemurro

Form HTML: Tipo radio del Tag input

radio: tipo del tag `input` per inserire un pulsante di opzione che consente di effettuare una sola scelta all'interno di un elenco di voci indicate con piccoli cerchi . L'eventuale elemento preselezionato è indicato con l'attributo **checked** . Tutti gli oggetti dell'elenco devono avere lo stesso valore per l'attributo **name**.

Esempio

Scegli la carta di credito

Visa

Mastercard

Scegli la carta di credito


```
<input type="radio" name="carta" value="visa"
      checked>Visa<br>
<input type="radio" name="carta"
      value="mastercard">Mastercard<br>
```

Prof. Montemurro

Form HTML: Tipo radio del Tag input

Esempio: tag input, tipo radio, con e senza attributo value

```
<form action="test.html" method="get">
    Scegli la carta di credito<br>
    <input type="radio" name="carta" checked>Visa<br>
    <input type="radio" name="carta"
           value="mastercard">Mastercard<br><br>
    <input type="submit" name="invia" value="Invia">
</form>
```

(i) File file:///C:/Users/Raffaele/Desktop/test.html?carta=on&invia=Invia

Scegli la carta di credito
 Visa
 Mastercard

Invia

Prof. Montemurro

Form HTML: Tipo radio del Tag input

Esempio: tag input, tipo radio, con e senza attributo value

(i) File file:///C:/Users/Raffaele/Desktop/test.html?carta=on&invia=Invia

Nota bene: il valore on del query parameter non è significativo.

Query parameter: coppia chiave-valore dove la chiave è il valore assegnato all'attributo name, mentre il valore è il valore assegnato all'attributo value.

Prof. Montemurro

Form HTML: Tipo radio del Tag input

continuazione esempio: tag input, tipo radio, con e senza attributo value

```
<form action="test.html" method="get">
    Scegli la carta di credito<br>
    <input type="radio" name="carta" checked>Visa<br>
    <input type="radio" name="carta"
           value="mastercard">Mastercard<br><br>
    <input type="submit" name="invia" value="Invia">
</form>
```

(i) File file:///C:/Users/Raffaele/Desktop/test.html?carta=mastercard&invia=Invia|

Scegli la carta di credito

Visa

Mastercard

Nota bene: il valore "mastercard" del query parameter è significativo (fa capire al server cosa gli è stato inviato).

Invia

Prof. Montemurro

Form HTML: Tipo radio del Tag input

Errore comune: i pulsanti di opzione radio appartenenti ad uno stesso elenco devono avere lo stesso name, ma il valore del name deve essere unico per ciascun elenco di pulsanti. **Il copia ed incolla fa dimenticare di cambiare il name.**

Esempio

Scegli la carta di credito

<input type="radio" name="carta" value="visa">Visa

<input type="radio" name="carta"
 value="mastercard">Mastercard

Selezione sesso

<input type="radio" name="sesso" value="M">Maschio

<input type="radio" name="sesso" value="F">Femmina

Prof. Montemurro

Form HTML: Tipo checkbox del Tag input

checkbox: tipo del tag `input` per inserire una casella di controllo utilizzata per selezionare anche più di una voce contemporaneamente in un elenco di voci indicate con piccoli quadrati . L'eventuale elemento preselezionato è indicato con l'attributo `checked` .

Esempio

Lingue conosciute


```
<input type="checkbox" name="cb1" value="ita"
       checked>italiano<br>
<input type="checkbox" name="cb2"
       value="ing">inglese<br>
<input type="checkbox" name="cb3"
       value="fra">francese<br>
```

Lingue conosciute
 italiano
 inglese
 francese

Prof. Montemurro

Form HTML: Tipo checkbox del Tag input

Esempio: tag `input`, tipo `radio`, con e senza attributo `value`

```
<form action="test.html" method="get">
  Scegli la carta di credito<br>
  Lingue conosciute<br>
  <input type="checkbox" name="cb1"
         checked>italiano<br>
  <input type="checkbox" name="cb2"
         value="ing">inglese<br>
  <input type="checkbox" name="cb3"
         value="fra">francese<br>
  <input type="submit" name="invia" value="Invia">
</form>
```

<input checked="" name="cb1" type="checkbox" value="ita"/> italiano	<input name="cb2" type="checkbox" value="ing"/> inglese
<input name="cb3" type="checkbox" value="fra"/> francese	<input type="button" value="Invia"/>

(C) File file:///C:/Users/Raffaele/Desktop/test.html?cb1=on&cb3=fra&invia=Invia

Prof. Montemurro

Form HTML: Tipo email del Tag input

email: tipo del tag `input` per inserire una o più e-mail; per inserire più email si usa l'attributo **multiple**.

Esempio 1: inserire una sola e-mail

E-mail <input type="email" name="email">

E-mail

Esempio 2: inserire più e-mail separate dalla virgola (senza spazi)

E-mails <input type="email" name="emails" multiple>

E-mails

Prof. Montemurro

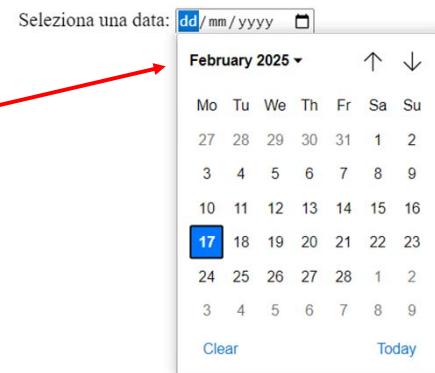
Form HTML: Tipo date del Tag input

date: tipo del tag `input` per inserire un selettore di date. Gli attributi **min** e **max** servono per indicare un intervallo di date valido.

Esempio 1

<label for="dataX">Seleziona una data:</label>
<input type="date" name="dataX" />

Seleziona una data: dd/mm/yyyy



Prof. Montemurro

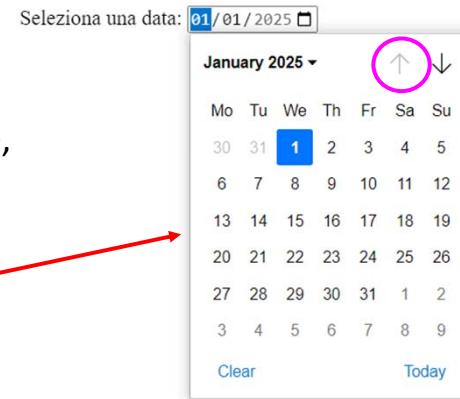
Form HTML: Tipo date del Tag input

Esempio 2

```
<label for="dataX">Seleziona una data:</label>
<input type="date" name="dataX" min="2025-01-01"
       max="2025-12-31">
```

Nota bene: freccia su è grigia per cui non si può scegliere una data antecedente allo 01/01/2025 (se andiamo al mese di dicembre, la freccia giù diventerà grigia perché non si può scegliere una data dopo il 31/12/2025).

Seleziona una data:



Prof. Montemurro

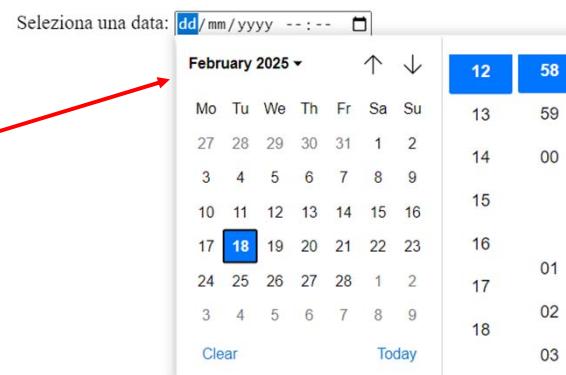
Form HTML: Tipo datetime-local del Tag input

datetime-local: tipo del tag input per inserire un selettore di date e di ora.

Esempio

```
<label for="dataX">Seleziona una data:</label>
<input type="datetime-local" name="dataX">
```

Seleziona una data:



Prof. Montemurro

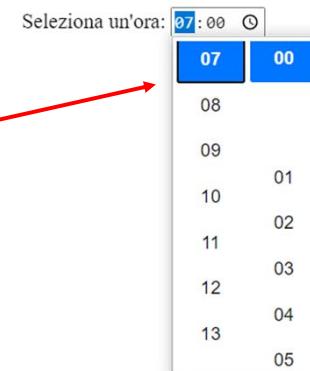
Form HTML: Tipo time del Tag input

time: tipo del tag input per inserire un'ora (ore e minuti, e optionalmente secondi); gli attributi **min** e **max** servono per indicare un intervallo di tempo valido).

Esempio

```
<label for="timeX">Seleziona un'ora:</label>
<input type="time" name="timeX">
```

Seleziona un'ora:



Prof. Montemurro

Form HTML: Tipo time del Tag input

Esempio 2

```
<label for="appt">Scegli l'ora della riunione:</label>
<input type="time" id="appt" name="appt" min="09:00"
       max="18:00">
<input type="submit" name="invia" value="Invia">
```

Scegli l'ora della riunione:

Value must be 09:00 or later.

Prof. Montemurro

Form HTML: Tipo url del Tag input

url: tipo del tag `input` per inserire un URL in un form.

Esempio

```
<label for="sito">Sito Web</label>
<input type="url" name="sito">
```

Sito Web

Prof. Montemurro

Form HTML: Tipo number del Tag input

number: per inserire un valore numerico; gli attributi **min** e **max** servono per specificare l'intervallo di valori validi. Sulla parte destra del campo appaiono due pulsanti per incrementare/decrementare il valore numerico.

Scegli l'età (14-40):

Esempio

```
<label for="eta">Scegli l'età (14-40):</label>
<input type="number" name="eta" min="14" max="40">
```

Prof. Montemurro

Form HTML: Tipo tel del Tag input

tel: per inserire un numero di telefono in un form; l'attributo **pattern** serve per specificare il formato del numero di telefono.

Esempio

```
<label for="phone">Numero di telefono:</label>
<input type="tel" name="tel" placeholder="123-456-
    7890" pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}">
<small>Formato: 123-456-7890</small>
```

Numero di telefono: Formato: 123-456-7890

Prof. Montemurro

Form HTML: Tipo range del Tag input

range: per visualizzare un **selettore** (o **slider**) che permette di scegliere un valore compreso in un intervallo. Gli attributi **min** e **max** servono per specificare l'intervallo dei valori. L'attributo **step** serve per specificare il passo tra un valore ed il successivo/precedente (es. **step="2"** significa che si passa da 0 a 2, da 2 a 4 ecc.).

Esempio per selezionare una percentuale nell'intervallo (0, 100) attraverso un selettore

```
0% <input type="range" name="perc" min=0 max=100
    step="1" value="15"> 100%
    0%  100%
```

Prof. Montemurro

Form HTML: Tipo color del Tag input

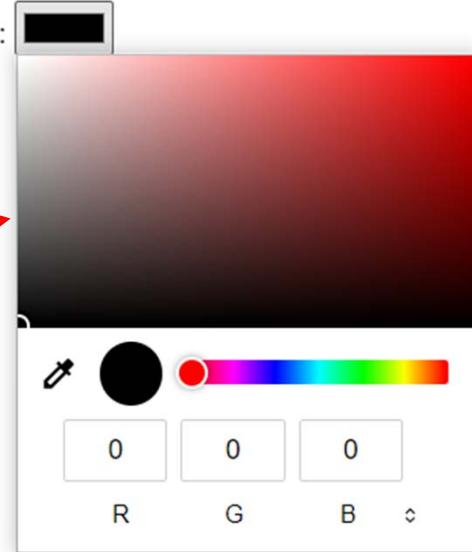
color: per selezionare un colore.

Esempio

```
<label>Colore preferito:</label>
<input type="color" name="color">
```

Colore preferito: 

Colore preferito: 



Prof. Montemurro

Form HTML: Tipo file del Tag input

file: per caricare uno o più file da inviare al server; per caricare più file si usa l'attributo **multiple**.

Esempio 1

```
<label for="myfile">Scegli un file:</label>
<input type="file" id="myfile" name="myfile">
```

Scegli un file: No file chosen

Esempio 2

```
<label for="myfile">Scegli i file:</label>
<input type="file" id="myfile" name="myfile" multiple>
```

Scegli i file: No file chosen

Prof. Montemurro

Form HTML: Tag button del Tag form e non solo

button: tag e attributo per inserire un pulsante cliccabile diverso dal pulsante di comando realizzato con il tipo button del tag `<input>` perché il contenuto del bottone può essere non solo un testo ma anche un'immagine.

Anche il tag `<button>` possiede le opzioni `submit` (pulsante di invio dei dati) e `reset` (pulsante di cancellazione dei dati) per l'attributo `type`, con lo stesso significato visto in precedenza per il tag `<input>`.

Esempio 1

```
<button type="button">fai clic qui</button> fai clic qui
<!-- tag button con type="button" in genere viene
usato con JavaScript (non lo vedremo) --&gt;</pre>

```

Prof. Montemurro

Form HTML: Tag button del Tag form

Esempio 2

```
<button type="button">
    
    <br>
    ...please
</button>
```



alt="fraseAlternativa" serve o per:

- i. mostrare la frase scritta quando il browser non riesce a visualizzare correttamente l'immagine;
- ii. far apparire un commento quando il mouse passa sopra l'immagine.

Prof. Montemurro

Form HTML: Tag textarea del Tag form

textarea: tag che permette di creare un'area nella quale inserire testo, specificando il numero di righe (attributo `rows`) e di colonne (attributo `cols`).

Esempio

```
<textarea name="richiesta" rows="3" cols="40">
    scrivi qui la tua domanda</textarea>
```

più comodo col **placeholder** (autocancellazione `testo` quando utente scrive):

```
<textarea name="richiesta" rows="3" cols="40"
placeholder="scrivi qui la tua domanda "></textarea>
```



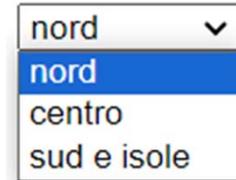
Prof. Montemurro

Form HTML: Tag select del Tag form

select: tag che permette di inserire un menù a tendina dove ciascuna voce è specificata con il tag **option**. L'attributo **selected** del tag option specifica un'eventuale voce preselezionata. L'attributo **multiple** del tag select consente all'utente di effettuare scelte multiple all'interno del menù a tendina (si tiene premuto tasto Ctrl per selezionare più voci).

Esempio

```
<form>
<select name="regione">
    <option value="1" selected>nord</option>
    <option value="2">centro</option>
    <option value="3">sud e isole</option>
</select>
</form>
```



Prof. Montemurro

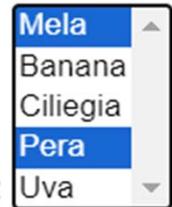
Form HTML: Tag select del Tag form

Esempio con scelte multiple nel menù a tendina

```
<form>
<label for="scelte">Scegli le tue opzioni:</label>
<select multiple name="scelte" id="scelte" size="5">
    <option value="mela">Mela</option>
    <option value="banana">Banana</option>
    <option value="ciliegia">Ciliegia</option>
    <option value="pera">Pera</option>
    <option value="uva">Uva</option>
</select>
</form>
```

Prof. Montemurro

Scegli le tue opzioni:



Form HTML: Tag fieldset del Tag form

fieldset: tag per indicare un insieme di campi, cioè un gruppo di dati collegati logicamente. Il tag **legend**, all'interno del tag **fieldset**, definisce il titolo del riquadro del **fieldset**. Il gruppo può contenere tutti i tipi di tag visti in precedenza.

Esempio

```
<fieldset>
    <legend>Dati del film</legend>
    <p><label for="titolo">Titolo</label>
        <input type="text" name="titolo" size="40"></p>
    <p><label for="regista">Regista</label>
        <input type="text" name="regista" size="30"></p>
    <p><label for="anno">Anno</label>
        <input type="text" name="anno"></p>
</fieldset>
```

Dati del film

Titolo	<input type="text"/>
Regista	<input type="text"/>
Anno	<input type="text"/>

Prof. Montemurro

Validazione Lato Client dei Dati di un Form

Alcuni tipi del tag di `input` permettono la **validazione lato client**, cioè permettono di controllare se i dati inseriti dall'utente sono validi prima di inviarli al server. Tali tag sono:

- [email](#)
- [url](#)
- [number](#)
- [tel](#)
- [date](#)
- [datetime-local](#)
- [range](#)

Nota bene: la validazione lato client non basta, serve sempre fare anche la validazione lato server perché l'utente potrebbe aver disabilitato quella lato client (noi non vedremo come fare la validazione lato server).

Prof. Montemurro

Validazione Lato Client dei Dati di un Form

Esempi: se il dato non è valido, cliccando su invia compare un messaggio

The image contains four separate screenshots of a web form demonstrating client-side validation:

- E-mail:** An input field contains "pippo.com". A validation message box says: "Please include an '@' in the email address. 'pippo.com' is missing an '@'."
- Sito Web:** An input field contains "sito@". A validation message box says: "Please enter a URL."
- Scegli l'età (14-40):** A dropdown menu shows "-1". A validation message box says: "Value must be greater than or equal to 14."
- Numero di telefono:** An input field contains "df". A validation message box says: "Please match the requested format."
- Numero di telefono:** An input field contains "12345-6". A validation message box says: "Please match the requested format."

Prof. Montemurro

Validazione Lato Client dei Dati di un Form

required: attributo booleano che serve per rendere obbligatoria la compilazione di un campo; se non si inserisce il dato richiesto, quando si clicca sul pulsante di invio dei dati, compare il messaggio "*Please fill in this field.*" (o "*Compila questo campo.*", o altro). L'attributo **required** si può usare con tutti i tag **input** del form.

Esempi

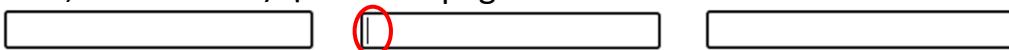
The screenshot shows a web form with several input fields and validation messages:

- Name:** An input field with the placeholder "Nome". Below it is a message: **Please fill in this field.**
- Email:** An input field with the placeholder "E-mail". Below it is a message: **Please fill in this field.**
- Date:** A date input field with the placeholder "Seleziona una data: dd/mm/yyyy". Below it is a message: **Please fill in this field.**
- Web Site:** An input field with the placeholder "Sito Web". Below it is a message: **Please fill in this field.**
- Age:** A dropdown menu with the placeholder "Scegli l'età (14-40)". Below it is a message: **Please fill in this field.**
- Phone Number:** An input field with the placeholder "Numero di telefono: 123-456-7890". Below it is a message: **Please fill in this field.**
- Format:** A text input field with the placeholder "Formato: 123-456-7890".

Prof. Montemurro

Form HTML: Attributo autofocus

autofocus: attributo booleano che serve per ottenere automaticamente il focus (cioè il cursore lampeggiante) di un elemento HTML (button, input, select, textarea) quando la pagina web viene caricata.



1. L'attributo **autofocus** non può funzionare con su più elementi HTML. Se applicato a più elementi, il primo in cui compare l'attributo **autofocus** riceverà il focus.
2. La messa a fuoco automatica di un controllo di un form può confondere le persone ipovedenti che utilizzano la tecnologia di lettura dello schermo e le persone con disabilità cognitive. Quando viene assegnata la messa a fuoco automatica, gli screen reader "teletrasportano" l'utente all'elemento HTML con l'**autofocus** senza avvisarlo in anticipo. La messa a fuoco automatica su un elemento può causare lo scorrimento della pagina durante il caricamento.

Prof. Montemurro

Form HTML: Attributo autofocus

Esempio 1: autofocus sulla casella di testo Nome

```
<label for="nome">Nome:</label>
<input type="text" id="nome" name="nome"
       autofocus><br><br>
<label for="cognome">Cognome:</label>
<input type="text" id="cognome"
       name="cognome"><br><br>
```

Casella Nome ottiene il focus appena
termina il caricamento della pagina web.

Nome:

Cognome:

Prof. Montemurro

Form HTML: Attributo autofocus

Esempio 2: autofocus sulla casella di testo Cognome

```
<label for="nome">Nome:</label>
<input type="text" id="nome" name="nome"><br><br>
<label for="cognome">Cognome:</label>
<input type="text" id="cognome" name="cognome"
       autofocus><br><br>
```

Casella Cognome ottiene il focus appena
termina il caricamento della pagina web.

Nome:

Cognome:

Prof. Montemurro

Differenza tra Attributo id e name

1. **id**: attributo opzionale che identifica univocamente un qualsiasi elemento HTML (sia elementi HTML del form che non); tale attributo ha i seguenti tre utilizzi:
 - i. identificare una sezione di una pagina HTML a cui poi si salterà tramite un link (vedi slide sui link: tag a con attributo href e identificatore di frammento);
 - ii. utilizzare uno specifico stile CSS per un elemento HTML (lo vedremo più avanti);
 - iii. accedere e modificare uno specifico elemento HTML tramite JavaScript (non lo vedremo).

Prof. Montemurro

Differenza tra Attributo id e name

2. **name**: attributo per assegnare un nome anche non univoco (es. ogni tipo radio del tag input appartenente allo stesso elenco ha lo stesso name) ad un elemento HTML. Tale attributo ha i seguenti utilizzi:
 - i. poter elaborare i dati inviati al server tramite un form HTML per cui è obbligatorio definirlo per gli elementi del form, infatti il suo valore è la chiave della coppia chiave-valore del query parameter;
 - ii. specificare il tipo di metadato da inserire col tag meta;
 - iii. ...

Prof. Montemurro



Prof. Montemurro

PHP (PHP Hypertext Preprocessor)

Classe 5ASI
ITCG Fermi

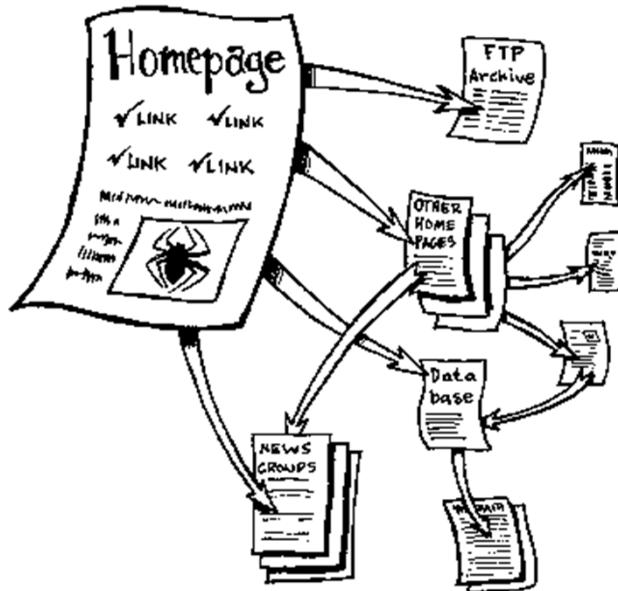
Prof. Montemurro

Linguaggi per Realizzare Pagine Web

1. Il linguaggio **HTML (HyperText Markup Language, o linguaggio di contrassegno per gli ipertesti)** consente la creazione di pagine web attraverso **file di testo** che sono interpretati dal browser che le visualizza in **forma grafica**. È un linguaggio per la gestione di ipertesti; **non** è un linguaggio di programmazione in quanto non viene utilizzato per svolgere elaborazioni, ma è un **linguaggio di formattazione della pagina**.
Ipertesto: testo multidimensionale (iper-), cioè a più dimensioni, infatti è costituito da un insieme di documenti che hanno tra loro un nesso logico in quanto si riferiscono ad una stessa classe di argomenti che possono essere consultati in modo non lineare mediante link a documenti esterni (**hyperlink**) per associazione di idee o di termini.
Navigazione ipertestuale: modalità di lettura dei documenti che prevede il salto da uno all'altro tramite i link.

Prof. Montemurro

Linguaggi per Realizzare Pagine Web



Prof. Montemurro

Traduttori: Compilatori ed Interpreti

- 1. Compilatori:** programmi che prendono in ingresso l'intero programma (file sorgente/i), e che restituiscono in uscita la rappresentazione dell'intero programma in linguaggio macchina (o codice binario 01...).
- 2. Interpreti:** programmi che traducono (in linguaggio macchina) ed eseguono direttamente ciascuna istruzione del programma/script scritto nel file sorgente/i, istruzione per istruzione.

Prof. Montemurro

Linguaggi per Realizzare Pagine Web

2. Il linguaggio **JavaScript JS** è:

- i. un linguaggio interpretato in quanto il codice non viene compilato, ma viene eseguito istruzione per istruzione direttamente dal browser;
- ii. un linguaggio di scripting (quindi è un linguaggio di programmazione) per cui è meglio parlare di script e non di programma.

Script: testo contenente una sequenza di istruzioni che può essere innestata anche nel codice di altri linguaggi, ed è eseguita da un programma interprete.

Limite dell'HTML e di JavaScript: non è possibile realizzare pagine web che consentano all'utente di accedere ai dati contenuti in un database che si trova su un server web.

Prof. Montemurro

Linguaggi per Realizzare Pagine Web

Linguaggi lato client (HTML, JavaScript ...): interpretazione delle istruzioni viene fatta direttamente dal browser web nel computer dell'utente che svolge il ruolo di client (richiedente).

Linguaggi lato server (PHP ...): interpretazione delle istruzioni avviene sul server, ed il risultato dell'elaborazione viene inviato al browser; le pagine web (dinamiche) contengono quindi il risultato del codice eseguito sul server.

Pagina web dinamica: pagina web che viene costruita dal server quando l'utente la richiede. A differenza di una pagina statica, che è un file HTML predefinito e identico per tutti gli utenti, una pagina dinamica può variare in base a diversi fattori, quindi è interattiva e personalizzabile.

Prof. Montemurro

Linguaggi per Realizzare Pagine Web

Server web (es. Apache): software per la gestione dei servizi web di un computer host o di un server di rete; tali servizi si basano sui protocolli standard delle reti e di internet, in particolare sul protocollo HTTP (HyperText Transfer Protocol) che è il principale protocollo informatico che consente il trasferimento di dati da e verso pagine web.

Il **PHP** (acronimo ricorsivo **PHP Hypertext Preprocessor, o preprocessore di ipertesti PHP**) è un linguaggio che estende le funzionalità del server web in quanto consente l'interpretazione di file con estensione *.php* contenenti il codice dell'applicazione lato server, oltre che naturalmente l'interpretazione dei classici marcatori (o tag) dei file con estensione *.html*.

Prof. Montemurro

Linguaggi per Realizzare Pagine Web

Il PHP è:

1. un linguaggio interpretato in quanto il codice non viene compilato, ma viene eseguito istruzione per istruzione direttamente dal server; l'**interprete PHP** viene aggiunto al server web come modulo esterno;
2. un linguaggio di scripting (quindi è un linguaggio di programmazione) in quanto permette di innestare gli script all'interno delle pagine web dove sono presenti i marcatori del linguaggio HTML. In questo modo si possono realizzare in modo dinamico **pagine web lato server** che non dipendono solo dai marcatori statici del linguaggio HTML, ma anche dalle azioni degli script realizzati tramite il linguaggio PHP;
3. un linguaggio integrato nel linguaggio HTML;
4. un linguaggio **general purpose**, cioè multiuso, in quanto non ha un utilizzo specifico, ma **generale**, anche se oggi viene usato solo per creare siti web dinamici.

Prof. Montemurro

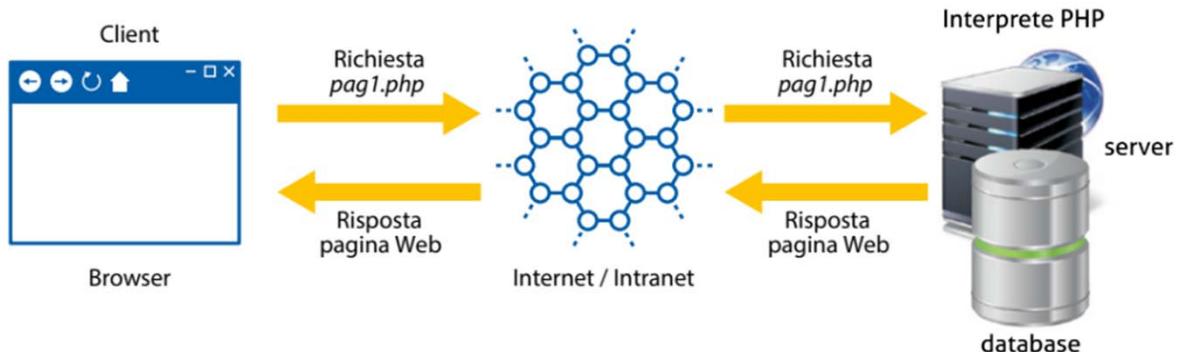
Linguaggi per Realizzare Pagine Web

Fasi eseguite dal server web quando viene richiesta una pagina con estensione `.php`:

1. legge il file di testo `.php` riga per riga;
2. quando trova i marcatori HTML li spedisce al browser;
3. quando trova i blocchi di codice HTML:
 - i. ne esegue l'interpretazione;
 - ii. recupera gli eventuali dati richiesti prelevandoli dai file o dai database del server;
 - iii. restituisce una pagina web, creata quindi in modo dinamico, visualizzabile dal browser.

Prof. Montemurro

Linguaggi per Realizzare Pagine Web



Prof. Montemurro

Linguaggi per Realizzare Pagine Web

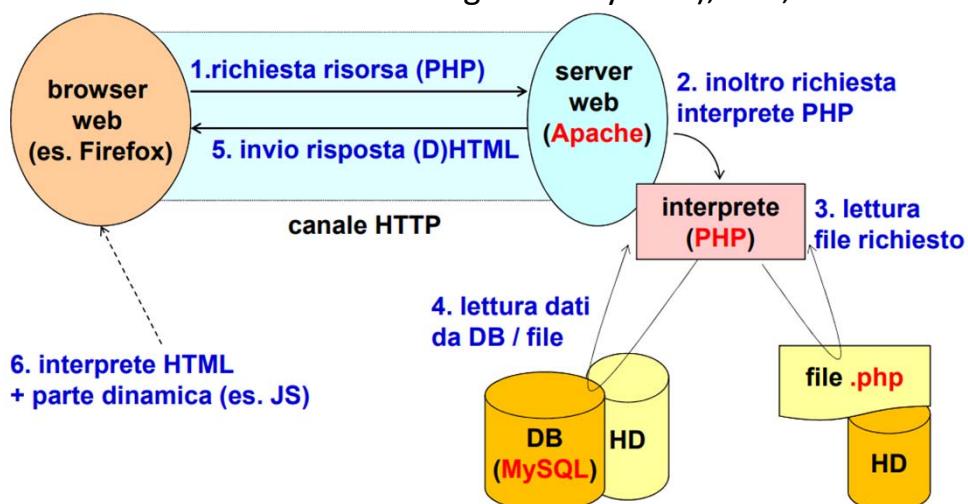
Vantaggi del PHP:

1. rende più veloce la creazione e lo sviluppo di pagine web;
2. facilita le fasi di manutenzione e di aggiornamento delle applicazioni;
3. i suoi script sono compatibili con diverse applicazioni;
4. include la possibilità di accedere a vari tipi di database;
5. è un software libero, e viene continuamente controllato ed aggiornato; è scaricabile dal sito internet www.php.net.

Prof. Montemurro

Ambiente di Sviluppo in Locale: XAMPP

XAMPP: Cross-platform, Apache (server web locale), MariaDB (MySQL che è un RDBMS Relational DataBase Management System), PHP, Perl



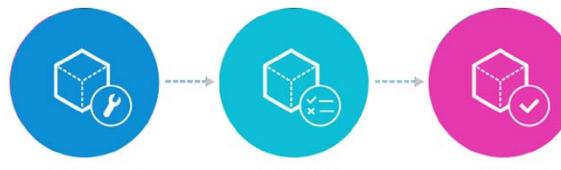
Prof. Montemurro

Ambiente di Sviluppo in Locale: XAMPP

XAMPP viene usato per sviluppare e testare le pagine web sul proprio computer locale, non è adatto all'ambiente di produzione.

Ambienti necessari per mettere in piedi un nuovo servizio o per eseguire cambiamenti significativi:

1. **ambiente di sviluppo** dove il servizio prende forma grazie agli sviluppatori
2. **ambiente di collaudo (o ambiente di test)** dove il servizio viene testato da chi ha i permessi
3. **ambiente di produzione** dove il servizio viene usato dai clienti



Prof. Montemurro

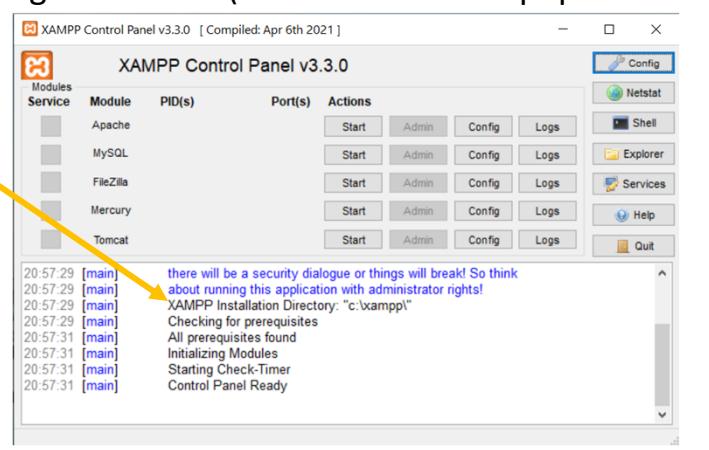
Ambiente di Sviluppo in Locale: XAMPP

Download URL: <https://www.apachefriends.org/download.html>

Percorso assoluto in cui creare i progetti XAMPP (es cartella con file php da testare): **C:\xampp\htdocs**

Nel mio caso ho deciso di creare la cartella progetti per motivi di ordine:

C:\xampp\htdocs\progetti



Prof. Montemurro

Ambiente di Sviluppo in Locale: XAMPP

Funzione **`phpinfo()`** restituisce in uscita le informazioni sull'interprete PHP installato nel computer (versione, caratteristiche del server web, tutte le configurazioni di PHP). Per usarla occorre creare un file di prova, chiamato ad esempio "*informazioni*" usando un editor di testi (es. blocco note) e copiare il seguente codice:

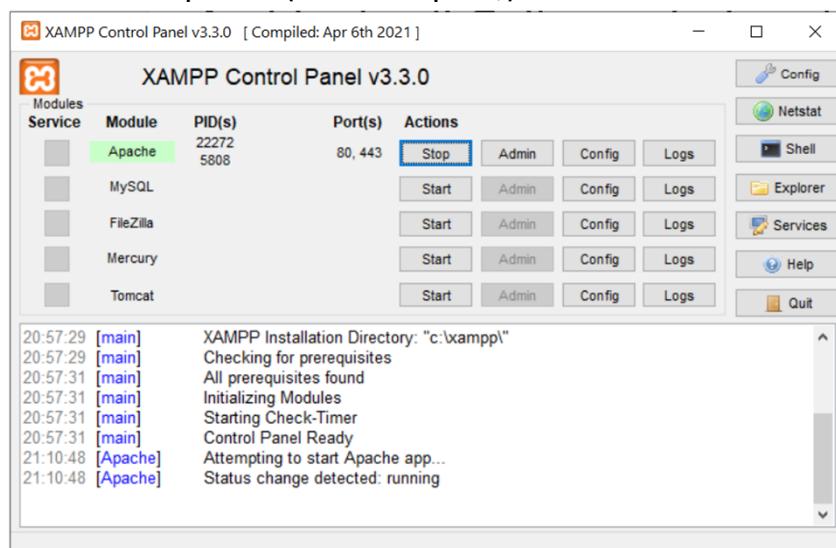
```
<?php
    phpinfo();
?>
```

una volta salvato, occorre cambiare l'estensione del file da *informazioni.txt* a *informazioni.php*. Questo andrà copiato nella cartella
C:\xampp\htdocs\progetti

Prof. Montemurro

Ambiente di Sviluppo in Locale: XAMPP

Avvio del server web Apache (80 listen port;)



Prof. Montemurro

Ambiente di Sviluppo in Locale: XAMPP

Per richiamare il file *informazioni.php* dal browser, occorre avviare Apache in XAMPP ed andare all'indirizzo <http://localhost/progetti/>



The screenshot shows the PHP version 8.2.12 information page. It includes the PHP logo at the top right and a table of system configuration details. The table rows are:

System	Windows NT PC-RAFFAELE 10.0 build 19045 (Windows 10) AMD64
Build Date	Oct 24 2023 21:10:40
Build System	Microsoft Windows Server 2019 Datacenter [10.0.17763]
Compiler	Visual C++ 2019
Architecture	x64
Configure Command	cscript /nologo /e:jscript configure.js "–enable-snapshot-build" "–enable-debug-pack" "–with-pdo-oci=..\..\..\instantclient\ sdk\shared" "–with-oci8-19=..\..\..\instantclient\ sdk\shared" "–enable-object-out-dir=..\\obj" "–enable-com-dotnet=shared" "–without-analyzer" "–with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	no value
Loaded Configuration File	C:\xampp\php\php.ini

Prof. Montemurro

Caratteristiche delle Pagine PHP

Le istruzioni del linguaggio PHP vengono scritte all'interno di pagine web, cioè in un file di testo salvato con l'estensione .php, e sono racchiuse tra il marcitore (o tag) di inizio **<?php** ed il marcitore (o tag) di fine **?>**. Questi marcatori si trovano nella sezione *body* della pagina HTML.

```
<?php
    // elenco di istruzioni in PHP
    ...
    echo ...; // visualizza dati o messaggi nel browser
    ...
?>
```

Prof. Montemurro

Caratteristiche delle Pagine PHP

Buona norma: se un file contiene solo codice PHP, è consigliabile omettere il tag di chiusura ?> alla fine del file stesso. In questo modo si evita che vengano aggiunti accidentalmente **spazi vuoti** o **nuove righe dopo** il tag di chiusura PHP, il che potrebbe causare effetti indesiderati perché PHP avvierà la bufferizzazione dell'uscita (*slide seguente) anche quando il programmatore non ha alcuna intenzione di inviare alcuna uscita al buffer in quel punto dello script, perché il buffer di uscita è abilitato di default.

Prof. Montemurro

Caratteristiche delle Pagine PHP

(*) **Buffer PHP:** pezzo di memoria sul server per memorizzare temporaneamente contenuti che altrimenti verrebbero inviati al flusso di uscita (es. al client). Ciò fornisce un controllo sul quando l'uscita sarà presentata all'utente.

Bufferizzazione dell'uscita (o **controllo dell'uscita**, o **output buffering**, o **output control**): memorizzazione temporanea dell'uscita nel buffer interno prima che venga svuotato (invia e scartato) nel browser (in un contesto web) o nella shell (sulla riga di comando). Mentre la bufferizzazione dell'uscita è attiva, nessuna uscita viene inviata dallo script sul flusso di uscita in quanto questa viene memorizzata in un buffer interno.

Prof. Montemurro

Caratteristiche delle Pagine PHP

Utilità della bufferizzazione dell'uscita

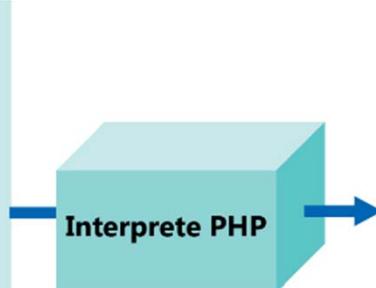
1. Le **reti** sono più **efficienti** quando inviano più dati in meno blocchi, piuttosto che meno dati in più blocchi.
Buona norma: consegnare il contenuto al browser web del client in meno pezzi per ridurre il numero totale di richieste HTTP.
Ecco perché si abilita la buffering di output PHP; il buffer di uscita di PHP raccoglie fino a 4.096 byte prima di riversarne il contenuto al client.
2. Quando c'è un ritardo nel reperire qualche dato da un database; il buffer di uscita aiuta a **prevenire** (i) **caricamenti parziali di pagina** o (ii) che **dati incompleti vengano mostrati agli utenti**.

Prof. Montemurro

Caratteristiche delle Pagine PHP

Pagina con codice PHP

```
<HTML>
  - -
<BODY>
  <?php
    //codice PHP
  ?>
</BODY>
</HTML>
```



Pagina ricevuta dal browser

```
<HTML>
  - -
<BODY>
  solo HTML
</BODY>
</HTML>
```

Prof. Montemurro

Commenti

```
<?php
:
// questo è un commento su una singola riga
:
/* questo è un
commento su più righe */
:
?>
```

I commenti non vengono presi in considerazione dall'interprete PHP.

Prof. Montemurro

Variabili

1. Le variabili in PHP sono identificate da un nome preceduto dal simbolo del dollaro \$.
2. I nomi delle variabili sono **case sensitive** (\$nome diverso da \$Nome).
3. I nomi delle variabili devono iniziare con una lettera.
4. I nomi delle variabili possono contenere lettere, cifre, carattere di sottolineatura _.
5. La dichiarazione delle variabili è implicita; non è necessario dichiarare le variabili in quanto una variabile viene creata nel momento in cui le si assegna un valore.
6. Il tipo di una variabile viene attribuito dal PHP in modo autonomo (non occorre specificarlo) e dinamico (può variare all'interno dello script; es.: la stessa variabile viene considerata di tipo numerico se le si assegna un numero, oppure viene considerata una stringa se le si assegna una stringa); ecco perché PHP è un **linguaggio dinamicamente tipizzato**.

Prof. Montemurro

Esempio Variabili

Il seguente codice assegna un valore a tre variabili; la dichiarazione delle tre variabili è implicita ed avviene quando si assegna loro un valore.

```
<?php
    $eta = 19;                                // intero
    $titolo = "I promessi sposi";              // stringa
    $prezzo = 31.2;                            // floating point
?>
```

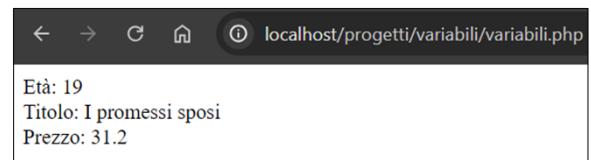
Prof. Montemurro

Esempio Variabili

Il seguente codice assegna un valore a tre variabili e le stampa a video; la dichiarazione delle tre variabili è implicita ed avviene quando si assegna loro un valore.

```
<?php
    $eta = 19;
    $titolo = "I promessi sposi";
    $prezzo = 31.2;
    echo "Età: $eta <br>";
    echo "Titolo: $titolo <br>";
    echo "Prezzo: $prezzo <br>";
?>
```

Il tag HTML `
` serve per andare a capo e viene interpretato dal browser.



Prof. Montemurro

Tipi di Variabili

Tipo di una variabile: specifica la **dimensione** (es. 1 byte) e la **natura** (es. booleano, carattere, numero) dell'informazione che sarà memorizzata in una variabile.

var_dump () : funzione predefinita PHP che stampa a video il tipo ed il valore della variabile che le viene passata come argomento.

In PHP ci sono:

1. **funzioni predefinite** (o **built-in function**), cioè già pronte all'uso;
2. **funzioni utente** (o **funzioni definite dall'utente**) (o **User-Defined Function UDF**) le quali non sono disponibili, ma vanno scritte.

Prof. Montemurro

Tipi di Variabili

1. **null**
2. **Numero intero**
3. **Numero a virgola mobile** (o **floating-point number**)
4. **Stringa**
5. **Valori logici** (o **valori booleani**)
6. **Array**
7. **Oggetto**
8. **callable**
9. **resource**

Nota bene: gli ultimi tre tipi non li vedremo.

Prof. Montemurro

Tipo null

null: è un tipo il cui unico valore costante è **null**; questo valore viene usato ogni volta che vogliamo rappresentare l'assenza di un valore da qualche parte (es. funzione che non ritorna alcun valore).

Una variabile è **null** nei seguenti tre casi:

1. le è stato assegnato il valore costante **null**;
2. non le è stato ancora assegnato alcun valore;
3. è stata cancellata con la funzione **unset()**.

Prof. Montemurro

Tipo null

Esempio 1: variabile null perché le è stato assegnato il valore costante null

```
$a = null;
var_dump($a);           // NULL (stampato a video)
```

Esempio 2: variabile null perché non le è stato ancora assegnato alcun valore

```
var_dump($b);           // NULL (stampato a video)
echo $b;                // non stampa alcunchè a video
```

Esempio 3: variabile null perché è stata cancellata con la funzione **unset()**

```
$a = 1;
var_dump($a);           // int(1) (stampato a video)
echo "<br>";
unset($a);              // NULL (stampato a video)
var_dump($a);
```

Prof. Montemurro

Tipo int

int: tipo numero intero

Esempio

```
$a = 1;  
var_dump($a); // int(1)
```

Prof. Montemurro

Tipo float

float: tipo numero a virgola mobile

Esempi

```
$a = 1.1;  
$b = 1.2e3; // 1.2 * 103  
$c = 7E-10; // 7 * 10-10  
var_dump($a, $b, $c); // float(1.1) float(1200)  
// float(7.0E-10)
```

Prof. Montemurro

Tipo float

Approfondimento

Curiosità: si parla di floating-point (punto fluttuante) perché, secondo la **notazione anglosassone**, la separazione tra parte intera e parte decimale è il **punto**, mentre nella **notazione italiana** è la **virgola**. I numeri a virgola mobile sono nati per rappresentare i numeri reali \mathbb{R} nei calcolatori; i numeri reali possono essere infiniti (es. π), mentre i numeri a virgola mobile sono un'approssimazione finita di \mathbb{R} basata su espansioni decimali finite.

Esempio: spostamento del punto (notazione anglosassone)

$$\begin{aligned} 0.1234 \times 10^{-2} &= 0.001234 \\ 0.1234 \times 10^1 &= 1.234 \\ 0.1234 \times 10^4 &= 1234 \\ 0.1234 \times 10^8 &= 12,340,000 \end{aligned}$$

Prof. Montemurro

Tipo string

Modi di assegnare una stringa ad una variabile:

1. racchiudendo il testo tra **doppi apici ";** questo metodo permette di espandere (es. stampare a video) eventuali altre variabili presenti all'interno della stringa;
2. racchiudendo il testo tra **apici singoli '**; tale metodo non consente di espandere eventuali altre variabili presenti all'interno della stringa.

Prof. Montemurro

Tipo string

Esempio: modi di assegnare una stringa ad una variabile

```
<?php
    $nome = "Giovanni";
    $saluto = "Buongiorno $nome";
    echo "$saluto <br>";
    $saluto = 'Buongiorno $nome';
    echo "$saluto <br>";
?>
```

Risultato nel browser:

Buongiorno Giovanni
Buongiorno \$nome

Prof. Montemurro

Tipo string

Sequenze di escape: sequenze di caratteri particolari che iniziano con la barra inversa (o backslash) \ e vengono interpretate, relativamente alla visualizzazione del codice PHP (non della pagina web) solo se racchiuse tra doppi apici:

- \n a capo
- \t tabulazione orizzontale
- \\\ barra inversa
- \t tabulazione
- \' apice
- \" virgolette
- \r ritorno carrello
- \\$ dollaro
- \a segnale acustico

Prof. Montemurro

Tipo string

Approfondimento

Esempio: differenza tra \n tra apici doppi, tra apici singoli, e


```
<?php
    echo "Ciao \n Pippo <br>";      → Ciao Pippo
    echo 'Ciao \n Pluto <br>';       → Ciao \n Pluto
    echo "Ciao <br> Pippo";          → Ciao
                                    Pippo
?>
```

Visualizzando il codice PHP con Ctrl + U, si vede che la sequenza di escape \n nel primo echo produce un ritorno a capo (cosa che non fa nella pagina web).

```
1 | Ciao
2 | Pippo <br>Ciao \n Pluto <br>Ciao <br> Pippo
```

Prof. Montemurro

Tipo bool

bool: tipo booleano, può assumere solo due valori, **true** o **false**.

Esempio

```
$a = true;
var_dump($a);           // bool(true)
```

Prof. Montemurro

Tipo array

Array in PHP: è una variabile che contiene un insieme di **valori** ciascuno dei quali è identificato da un **indice** (o **chiave**). Quindi un array è un insieme di elementi, e ciascun elemento è costituito da una coppia chiave-valore (o indice-valore).

In PHP gli indici dell'array possono essere:

1. interi; in questo caso si hanno **array numerici**; l'indice è un numero, quello associato al valore del primo elemento è 0;
2. stringhe; in questo caso si hanno **array associativi**; l'indice è una stringa tra apici semplici o doppi e viene associato con l'**operatore di associazione =>**;
3. misti (interi e stringhe); in questo caso si hanno **array misti**.

In genere si preferisce "indice" per 1, e "chiave" per 2 e 3.

Prof. Montemurro

Tipo array

Sintassi per definire il valore di un elemento dell'array:

```
$nomeArray[indice0] = valore0;
$nomeArray[indice1] = valore1; ...
```

oppure si usa la parola chiave **array** (vedi esempi nelle slide successive).

Sintassi per l'accesso al valore di un elemento dell'array:

```
$pippo = $nomeArray[indice]
```

Prof. Montemurro

Tipo array: Esempio 1 Array Numerico

Esempio 1: array numerico

```
// primo modo di definire un array numerico
$voti[0] = 8.0;
$voti[1] = 6.5;
$voti[2] = 5.5;
$voti[3] = 7.0;
```

oppure con la parola chiave **array**:

```
// secondo modo di definire un array numerico
      0   1   2   3
$voti = array(8.0, 6.5, 5.5, 7.0);
```

Prof. Montemurro

Tipo array: Esempio 1 Array Numerico

continuazione esempio 1: array numerico

```
// terzo modo di definire un array numerico
$voti[] = 8.0;
$voti[] = 6.5;
var_dump($voti); // array(2) { [0]=> float(8) [1]=>
                  float(6.5) }
```

Principio di funzionamento: quando si assegna un valore ad un elemento di un array, se non viene specificato alcun indice (cioè se si usa `[]`), viene preso il **massimo degli indici int** esistenti ed il nuovo indice sarà **quel** valore massimo più 1. Se non esiste ancora alcun indice int, l'indice sarà 0 (zero); ciò succede in due casi:

1. quando si crea un nuovo array assegnando un valore al primo elemento (vedi esempio sopra);
2. quando l'array già esiste, ma gli indici esistenti non sono int.

Prof. Montemurro

Tipo array: Esempio 2 Array Numerico

Nel linguaggio PHP, a differenza di altri linguaggi di programmazione, un array può essere costituito da elementi di tipo diverso tra loro, cioè non è necessario che siano tutti dello stesso tipo.

Esempio 2: array numerico

```
$dati[0] = 30;
$dati[1] = "Roma";
$dati[2] = 12.7;
$dati[3] = true;
$dati[4] = null;
// o equivalentemente
$dati = array(30, "Roma", 12.7, true, null);
```

Prof. Montemurro

Tipo array: Esempio 2 Array Numerico

...continuazione esempio 2: coppia chiave-valore per array numerico

0	1	2	3	4
---	---	---	---	---

```
$dati = array(30, "Roma", 12.7, true, null);
var_dump($dati);

/* Output var_dump($dati);
array(5) { [0]=> int(30) [1]=> string(4) "Roma"
           [2]=> float(12.7) [3]=> bool(true) [4]=> NULL}
*/
```

Numero di elementi dell'array

↑

Numero di caratteri della stringa Roma

Prof. Montemurro

Tipo array: Esempio Array Associativo

Esempio: array associativo

```
$persona["cognome"] = "Rossi";
$persona["nome"] = "Toto";
$persona["nascita"] = 2000;
$persona["professione"] = "impiegato";
```

oppure con la parola chiave **array** e l'operatore di associazione =>:

```
$persona = array("cognome"=>"Rossi", "nome"=>"Toto",
    "nascita"=>2000, "professione"=>"impiegato");
```

Prof. Montemurro

Tipo array: Esempio Array Associativo

...continuazione esempio: coppia chiave-valore per array associativo

```
$persona = array("cognome"=>"Rossi", "nome"=>"Toto",
    "nascita"=>2000, "professione"=>"impiegato");
var_dump($persona);

/* Output var_dump($prodotto);
array(4) { ["cognome"]=> string(5) "Rossi"
            ["nome"]=> string(4) "Toto"
            ["nascita"]=> int(2000)
            ["professione"]=> string(9) "impiegato" }
*/
```

Prof. Montemurro

Tipo array: Esempio Array Misto

Esempio: array misto

```
$prodotto[0] = "Gelato";
$prodotto[1] = "Torta";
$prodotto["buono"] = "Lasagne";
```

oppure con la parola chiave **array** e l'**operatore di associazione =>**:

```
$prodotto = array(0=>"Gelato", 1=>"Torta",
                    "buono"=>"Lasagne");
```

Prof. Montemurro

Tipo array: Esempio Array Misto

...continuazione esempio: coppia chiave-valore per array misto

```
$prodotto = array(0=>"Gelato", 1=>"Torta",
                    "buono"=>"Lasagne");
var_dump($prodotto);

/* Output var_dump($prodotto);
array(3) { [0]=> string(6) "Gelato"
          [1]=> string(5) "Torta"
          ["buono"]=> string(7) "Lasagne" }
*/
```

Prof. Montemurro

Conversioni Automatiche di Tipo

Conversioni automatiche (o **conversioni implicite**, o **cast impliciti**) di tipo riguardano le **espressioni logiche**:

1. ogni valore numerico diverso da 0 vale `true`, mentre lo 0 vale `false`;
2. una qualsiasi stringa non vuota vale `true`, mentre la stringa vuota, "", vale `false`;
3. valore `null` vale `false`.

Prof. Montemurro

Conversioni Automatiche di Tipo

Esempio 1: in espressioni logiche 0 vale `false`, ogni numero ≠ 0 vale `true`

```
$a = 0;
var_dump(!$a); // !$a è bool(true), dunque $a è false
echo "<br> <br>";
$b = -123.456;
var_dump(!$b); // !$b è bool(false), dunque $b è true
```

Prof. Montemurro

Conversioni Automatiche di Tipo

Esempio 2: in espressioni logiche stringa non vuota vale true, stringa vuota "" vale false

```
$a = "";
var_dump(!$a); // !$a è bool(true), dunque $a è false
echo "<br> <br>";
$b = "ciao";
var_dump(!$b); // !$b è bool(false), dunque $b è true
```

Prof. Montemurro

Conversioni Automatiche di Tipo

Esempio 3: in espressioni logiche null vale false

```
$a = null;
var_dump(!$a); // !$a è bool(true), dunque $a è false
```

Prof. Montemurro

Conversioni Automatiche di Tipo

Una **stringa contenente un valore numerico**, quando compare in un'espressione aritmetica, viene automaticamente convertita in un opportuno tipo numerico.

Esempi

```
$str = "3";
$str = $str * 2;           // $str è intero e vale 6

$val = 1;
$val = $val + "3";        // $valore vale 4
```

Prof. Montemurro

Conversioni Automatiche di Tipo

Esempi

```
$val = 5;
$val = $val + "3.4Volt"; // $valore è float e vale 8.4

$val = 5;
$val = $val + null;     // $valore vale 5
```

Prof. Montemurro

Conversioni Esplicite di Tipo

Conversioni esplicite (o **cast espliciti**) di tipo si eseguono anteponendo alla variabile il tipo che si vuole ottenere racchiuso tra parentesi tonde: (int), (float), (double), (bool), (string).

Esempio

```
$a = 1;
var_dump($a);           // int(1)
echo "<br> <br>";
$b = (float) $a;
var_dump($b);           // float(1)
echo "<br> <br>";
$c = (double) $a;
var_dump($c);           // float(1), no double(1)
```

Prof. Montemurro

Costanti

Costanti: il loro valore non può essere modificato dopo che sono state definite; per definire una costante si usa la funzione **define** la quale ha due ingressi separati da una virgola, il nome della costante ed il suo valore.

Convenzione: i nomi delle costanti sono scritti in maiuscolo.

Nota bene: le costanti sono utilizzate nel codice indicandone il nome senza il simbolo del dollaro \$ iniziale (es. slide successiva).

Prof. Montemurro

Costanti

Esempio

```
<?php
    define("PERCENTUALE", 20); // percentuale di sconto
    define("G", 9.8);           // costante di gravità
    define("REGIONE", "Marche"); // nome della regione

    // uso costante PERCENTUALE senza $
    echo PERCENTUALE;
?>
```

Prof. Montemurro

Operatori Aritmetici

Operatori aritmetici: si usano coi numeri e le variabili numeriche, e sono:

- + addizione
- sottrazione
- * moltiplicazione
- / divisione
- % resto della divisione intera
- ++ incremento unitario
- decremento unitario

Prof. Montemurro

Operatore di Concatenazione

Operatore di concatenazione . (punto): per concatenare le stringhe.

Esempio

```
<?php
    $nome = "Giovanni";
    $saluto = "Buongiorno " . $nome;
    echo $saluto . "<br>";           → Buongiorno Giovanni
    $somma = 768;
    $messaggio = "Totale " . $somma;   → Totale 768
    echo "$messaggio <br>";
?>
```

L'istruzione `echo $messaggio . "
"`; è equivalente all'ultima.

Prof. Montemurro

Operatori di Assegnamento

Operatore di assegnamento = per assegnare un valore ad una variabile (già visto).

Operatori di assegnamento combinati:

Operatore	Esempio	Equivale a
<code>+=</code>	<code>\$a += \$b</code>	<code>\$a = \$a + \$b</code>
<code>-=</code>	<code>\$a -= \$b</code>	<code>\$a = \$a - \$b</code>
<code>*=</code>	<code>\$a *= \$b</code>	<code>\$a = \$a * \$b</code>
<code>/=</code>	<code>\$a /= \$b</code>	<code>\$a = \$a / \$b</code>
<code>%=</code>	<code>\$a %= \$b</code>	<code>\$a = \$a % \$b</code>
<code>.=</code>	<code>\$a .= \$b</code>	<code>\$a = \$a . \$b</code>

Prof. Montemurro

Operatori di Confronto

Operatori di confronto per confrontare due valori:

Operatore	Esempio	Descrizione
<code>==</code>	<code>\$a == \$b</code>	true se \$a è uguale a \$b <u>dopo manipolazione del tipo</u> (cioè <u>conversione implicita</u>)
<code>===</code>	<code>\$a === \$b</code>	true se \$a è uguale a \$b, <u>e se \$a e \$b sono dello stesso tipo (no manipolazione tipo)</u>
<code>!==</code>	<code>\$a != \$b</code>	true se \$a è diverso da \$b <u>dopo manipolazione del tipo</u>
<code>!=</code>	<code>\$a !== \$b</code>	true se \$a <u>non</u> è uguale a \$b, <u>oppure</u> \$a e \$b <u>non sono dello stesso tipo (no manipolazione tipo)</u>
<code><</code>	<code>\$a < \$b</code>	true se \$a è minore di \$b
<code>></code>	<code>\$a > \$b</code>	true se \$a è maggiore di \$b
<code><=</code>	<code>\$a <= \$b</code>	true se \$a è minore o uguale a \$b
<code>>=</code>	<code>\$a >= \$b</code>	true se \$a è maggiore o uguale a \$b

Prof. Montemurro

Operatori di Confronto

Esempio

```
$a = 5;
$b = "5";
var_dump($a);           // int(5)
echo "<br> <br>";
var_dump($b);           // string(1)"5"
echo "<br> <br>";
var_dump($a == $b);     // bool(true) (*)
echo "<br> <br>";
var_dump($a === $b);    // bool(false)
//(*) conversione implicita di $a da int a bool, e
// di $b da string a bool
```

Prof. Montemurro

Operatori Logici

Operatori logici per valutare le espressioni logiche:

!	not
&&	and vero solo se entrambi gli operandi sono veri
	or vero se almeno uno degli operandi è vero
xor	xor vero se solo uno dei due operandi è vero

\$a	\$b	!\$a	\$a && \$b	\$a \$b	\$a xor \$b
true	true	false	true	true	false
true	false	false	false	true	true
false	true	true	false	true	true
false	false	true	false	false	false

Prof. Montemurro

Messaggi di Errore

Approfondimento

Se l'esecuzione dello script provoca **errori**, per scelta predefinita non vengono visualizzati messaggi di errore e compare una **pagina bianca** del browser.

Per conoscere il tipo di errore che è stato riscontrato, si deve consultare il file err.log (o **error.log**) nella cartella logs del server Web (Apache), dove gli errori vengono registrati in ordine cronologico con data, ora, tipo di errore e numero della linea dello script che ha provocato la situazione di errore

Prof. Montemurro

Messaggi di Errore

Approfondimento

Buona norma: in fase di collaudo conviene inserire le seguenti due istruzioni all'inizio dello script PHP per visualizzare gli eventuali errori generati durante l'esecuzione dello script stesso.

```
error_reporting(E_ALL);
ini_set("display_errors", 1);
```

1. La funzione **error_reporting**(livello_errore) serve per specificare quali tipi di errori devono essere visualizzati; passandole come argomento la costante **E_ALL**, la funzione visualizzerà tutti i tipi di errori.
2. In generale la funzione **ini_set**(opzione, valore) imposta un'opzione di configurazione; l'opzione **display_errors** stampa a video gli errori se il suo valore è 1, non li stampa se il suo valore è 0.

Prof. Montemurro

Messaggi di Errore

Approfondimento

Esempio

```
<?php
echo $a;
?>
```

Schermata bianca



```
<?php
error_reporting(E_ALL);
ini_set("display_errors", 1);
echo $a;
?>
```

Messaggio d'errore

Notice: Undefined variable: a in /home/rxNHHh/prog.php on line 4

Prof. Montemurro

Funzioni

Obiettivi

1. Suddividere un problema in sottoproblemi più semplici da risolvere (ogni sottoproblema è una funzione, dunque ha senso creare una funzione anche nel caso in cui dovesse essere chiamata una sola volta)
2. Ripetere più volte determinate procedure

Funzioni: permettono di isolare porzioni ben definite di codice, e di assegnare loro un nome che sarà usato per richiamarle nei punti del programma in cui servono.

Prof. Montemurro

Funzioni

In PHP, a differenza di C++, non è possibile dichiarare una funzione senza definirla (no prototipi), ma una funzione deve essere sempre dichiarata e definita. Per farlo:

1. si usa la parola chiave **function** seguita dal nome della funzione;
2. si specificano eventuali **parametri formali** (o **argomenti formali**) i quali sono racchiusi tra parentesi tonde dopo il nome della funzione, e sono separati dalla virgola;
3. le istruzioni della funzione sono racchiuse tra parentesi graffe; questo è il **corpo della funzione**;
4. se la funzione deve restituire un valore, si usa l'istruzione **return** seguita dal valore da restituire.

Prof. Montemurro

Funzioni

Sintassi per la **definizione** di una funzione definita dall'utente:

```
function nomeFunzione (pFormale1, pFormale2,...) {
    // istruzioni
    // eventuale return di qualcosa
}
```

Sintassi per la **chiamata** della funzione nomeFunzione:

```
nomeFunzione (pAttuale1, pAttuale2,...);
```

Chiaramente, se c'è un **return**, occorre salvare il valore in un contenitore:

```
nomeContenitore = nomeFunzione (pAttuale1, pAttuale2,...);
```

Prof. Montemurro

Funzioni

Esempio: funzione per calcolare l'area di un rettangolo

Parametri formali (o argomenti formali)

```
<?php
function areaRettangolo($base, $altezza) {
    $area = $base * $altezza;
    return $area;
}
```

Parametri attuali (o argomenti attuali)

```
$risultato = areaRettangolo(2, 3);
echo $risultato;                                // 6
?>
```

Prof. Montemurro

Funzioni

Modalità di passaggio dei parametri alle funzioni

	Passaggio per valore	Passaggio per riferimento (&)	Passaggio per valore predefinito
Modifica parametri attuali	No	Si	No

Prof. Montemurro

Funzioni

Esempio: passaggio di parametri per valore predefinito (o per default)

```
<?php
function stampa($param = "predefinito") {
    echo "Il parametro è: " . $param . "<br>";
}

stampa(); -----> Il parametro è: predefinito
stampa("come mi pare!");-----> Il parametro è: come mi pare!
?>
```

Prof. Montemurro

Funzioni

Esempio: passaggio di parametri per riferimento

```
<?php
function nominativo(&$nome) {
    echo $nome . "<br>";
    $nome = "Pluto";
}

$nomeOriginale = "Pippo";
nominativo($nomeOriginale);
echo $nomeOriginale; ?>
```

Prof. Montemurro

Funzioni

Scope di una variabile (o **ambito di una variabile**, o **visibilità di una variabile**): parte di programma in cui la variabile può essere usata.

Variabili locali: tutte le variabili che definiamo all'interno del corpo di una funzione.

Variabili globali: tutte le variabili definite fuori dalle funzioni.

Attenzione: i parametri formali e le variabili locali hanno senso solo all'interno della funzione stessa, cioè i parametri formali e le variabili locali sono visibili ed utilizzabili (nascono e muoiono) solo all'interno della funzione. Il loro scope è l'interno della funzione.

Prof. Montemurro

Funzioni

Esempio: scope delle variabili locali

```
<?php
function f1() {
    $variabile = 1;
    var_dump($variabile);
}

var_dump($variabile);
echo "<br>";
f1();
var_dump($variabile);
?>
```

Prof. Montemurro

Funzioni

Esempio: scope delle variabili globali

```
<?php
function f1() {
    var_dump($numero);
}

$numero = 3.14;
var_dump($numero);
echo "<br>";
f1();
?>
```

Prof. Montemurro

Funzioni: Parametri Tipizzati

Approfondimento

Nelle funzioni è possibile specificare di che tipo devono essere i suoi parametri.

Esempio

```
<?php
function areaRettangolo(int $base, int $altezza) {
    $area = $base * $altezza;
    return $area;
}

$risultato = areaRettangolo(2, "ciao");
// PHP Fatal error: Uncaught TypeError: Argument 2
passed to areaRettangolo() must be of the type
integer, string given...thrown...on line 2
?>
```

Prof. Montemurro

Funzioni: Parametri Tipizzati

Approfondimento

Esempio

```
<?php
function areaRettangolo(int $base, int $altezza) {
    $area = $base * $altezza;
    return $area;
}

$risultato = areaRettangolo(2, "3 ciao");
echo $risultato; // 6
// Attenzione: nonostante il secondo argomento attuale
non sia un intero, ma una stringa "3 ciao", si ha una
conversione implicita da stringa ad intero per cui
$altezza non vale "3 ciao", ma vale 3
?>
```

Prof. Montemurro

Funzioni: Parametri Tipizzati

Approfondimento

Per garantire che il tipo degli argomenti attuali e formali sia esattamente lo stesso (e quindi per evitare anche conversioni implicite), dopo il tag di apertura php si usa **declare(strict_types=1)** ;

Esempio

```
<?php declare(strict_types=1);
function areaRettangolo(int $base, int $altezza) {
    $area = $base * $altezza;
    return $area;
}
$risultato = areaRettangolo(2, "3 ciao");
// PHP Fatal error: Uncaught TypeError: Argument 2
passed to areaRettangolo() must be of the type integer,
string given...thrown...on line 2
?>
```

Prof. Montemurro

Funzioni: return Tipizzato

Approfondimento

Specificare tipo dell'eventuale valore di ritorno: (1) si scrive : nomeTipo tra la parentesi tonda di chiusura della lista degli argomenti e la parentesi graffa di apertura del corpo della funzione; (2) dopo il tag di apertura php si usa **declare(strict_types=1)** ; per evitare conversioni implicite.

Esempio

```
<?php // qui non ho scritto declare(strict_types=1);
function areaRettangolo(float $base, float $altezza) : int {
    $area = $base * $altezza;
    return $area;
}
$risultato = areaRettangolo(2.2, 3.0);
echo $risultato;           // 6
// Conversione implicita di $area = 2.2 * 3.0 = 6.6 da float
ad int, ecco perché il risultato è 6 e non 6.6
?>
```

Prof. Montemurro

Funzioni: return Tipizzato

Approfondimento

Esempio

```
<?php declare(strict_types=1);
function areaRettangolo(float $base, float $altezza) : int {
    $area = $base * $altezza;
    return $area;
}
$risultato = areaRettangolo(2.2, 3.0);
echo $risultato;
// PHP Fatal error: Uncaught TypeError: Return value of
areaRettangolo() must be of the type integer, float
returned...thrown...on line 4
?>
```

Possibili soluzioni: (1) scrivo : float o (2) scrivo return (int) \$area; per convertire esplicitamente il float in int.

Prof. Montemurro

Strutture di Controllo

Strutture di controllo: sono costrutti per controllare l'ordine di esecuzione delle istruzioni (i costrutti sono mattoncini da costruzione di un linguaggio di programmazione). Le due strutture di controllo che vedremo sono:

1. le **strutture di controllo di selezione** (if...else; if...elseif; switch...case);
2. le **strutture di controllo di iterazione (o di ripetizione)** (while; do...while; for; foreach).

Prof. Montemurro

1i Strutture di Controllo di Selezione: if...else

Sintassi generale della struttura di controllo di selezione if...else

```
if (condizione) {
    // istruzioni eseguite se la condizione è vera
} else {
    // istruzioni eseguite se la condizione è falsa
}
```

Se il blocco di istruzioni è composto da una sola istruzione, si possono omettere le parentesi graffe {}:

```
if (condizione) //istruz. eseguite se condizione è
vera
else // istruzioni eseguite se la condizione è falsa
```

Prof. Montemurro

1ii Strutture di Controllo di Selezione: if...elseif

Sintassi generale della struttura di controllo if...elseif (o if...else if) per la selezione annidata

```
$valore = ...;
if ($valore > 0)
    echo "valore maggiore di 0";
elseif ($valore == 0)
    echo "valore uguale a 0";
else
    echo "valore minore di 0";
```

Nota bene: si può usare sia la sintassi **if...elseif** che **if..else if.**

Prof. Montemurro

1iii Strutture di Controllo di Selezione: switch...case

Sintassi generale della struttura di controllo switch...case

```
switch (espressione) {
    case valore1:
        // istruzioni
        break;
    case valore2:
        // istruzioni
        break;
    // eventuali altri case
    default:
        // istruzioni; qui non serve break;
}
```

Prof. Montemurro

1iii Strutture di Controllo di Selezione: switch...case

switch...case: quando il programma arriva all'istruzione switch, viene valutata l'espressione tra parentesi tonde;

- se l'espressione è uguale al valore1, si esegue l'alternativa case valore1;
- se l'espressione è uguale al valore2, si esegue l'alternativa case valore2;
- ...in generale, si esegue l'alternativa corrispondente al case "che ha lo stesso valore" dell'espressione che compare accanto all'istruzione switch;
- l'alternativa **default** è opzionale e, se presente, viene eseguita solo se nessuna delle alternative precedenti è stata eseguita.

Se non c'è l'alternativa default e se il valore dell'espressione non coincide con alcun valore dei vari case, allora lo switch non viene eseguito e si passa all'istruzione successiva.

Prof. Montemurro

1iii Strutture di Controllo di Selezione: switch...case

Esempio: switch...case

```
$vocale = ...
switch($vocale) {
    case "A":
    case "a":
        echo "Hai scelto la vocale a.";
        ...
    case "U":
    case "u":
        echo "Hai scelto la vocale u.";
    default:
        echo "Non hai scelto alcuna vocale.";
}
```

Prof. Montemurro

2 Strutture di Controllo di Iterazione

Ciclo (o **struttura di controllo di iterazione**, o **struttura di controllo di ripetizione**): ripetizione di una o più istruzioni (in un algoritmo) un numero di volte definito o indefinito, specificato tramite una condizione; tale ripetizione è chiamata **iterazione**.

I cicli sono utili quando si deve operare con lunghe liste di dati, per esempio con gli array.

Tipi di strutture di controllo di iterazione

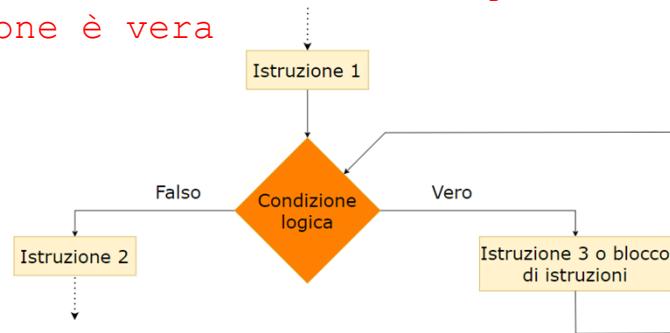
- i. ciclo while
- ii. ciclo do...while
- iii. ciclo for
- iv. ciclo foreach

Prof. Montemurro

2i Ciclo while

Sintassi generale del ciclo while (in italiano significa quando)

```
<?php
// istruzione 1
while (condizione) {
    // istruzione 3 o blocco di istruzioni eseguite fin
    // quando la condizione è vera
}
// istruzione 2
?>
```

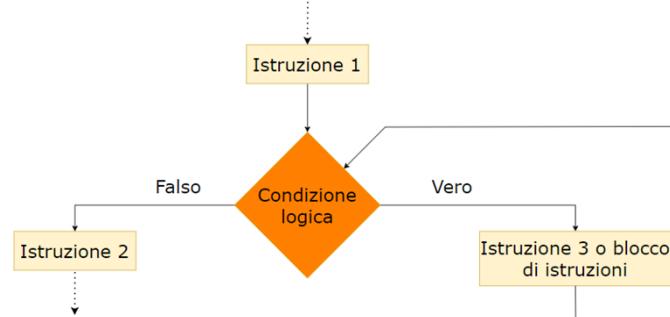


Prof. Montemurro

2i Ciclo while

while: quando nel programma si arriva all'istruzione while, viene valutata la condizione tra parentesi tonde;

- se la condizione vale true, viene eseguito il blocco di istruzioni tra parentesi graffe, cioè il **corpo del while**; dopo si ritorna alla condizione, e se vale di nuovo true, si ripete un'altra volta il corpo del while;
- se la condizione vale false, si esce dal ciclo while e si esegue l'istruzione che segue il ciclo while.



Prof. Montemurro

2i Ciclo while

Esempio

```
// Questo ciclo while stampa i numeri da 0 a 9
$i = 0;
while ($i < 10) {
    echo "Il valore di i è: " . $i . "<br>";
    $i++;
}
```

Output

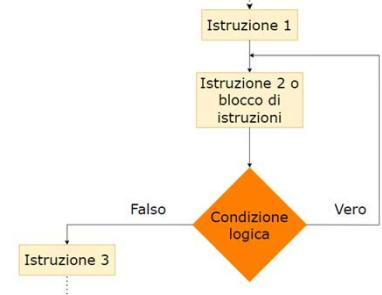
```
Il valore di i è: 0
Il valore di i è: 1
Il valore di i è: 2
Il valore di i è: 3
Il valore di i è: 4
Il valore di i è: 5
Il valore di i è: 6
Il valore di i è: 7
Il valore di i è: 8
Il valore di i è: 9
```

Prof. Montemurro

2ii Ciclo do...while

Sintassi generale del ciclo do...while (viene eseguito almeno una volta)

```
<?php
// istruzione 1
do {
    // istruzione 2 o blocco di istruzioni eseguite fin
    // quando la condizione è vera
} while (condizione);
// istruzione 3
?>
```

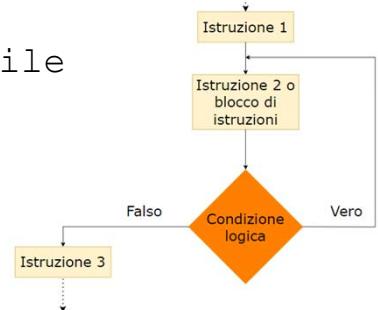


Prof. Montemurro

2ii Ciclo do...while

do...while: quando nel programma si arriva all'istruzione `do`, viene eseguito almeno una volta il blocco di istruzioni tra parentesi graffe, cioè il **corpo del do** (nel caso del ciclo `while`, il corpo del `while` viene eseguito per la prima volta solo se la condizione vale `true`). In seguito viene valutata la condizione tra parentesi tonde;

- se la condizione vale `true`, viene eseguito nuovamente il **corpo del do**; dopo si ritorna alla condizione, e se vale di nuovo `true`, si ripete un'altra volta il corpo del `do`;
- se la condizione vale `false`, si esce dal ciclo `do...while` e si esegue l'istruzione che segue il ciclo `do...while`.



Prof. Montemurro

2ii Ciclo do...while

Esempio

```
// Questo ciclo do...while stampa i numeri da 0 a 9
$i = 0;
do {
    echo "Il valore di i è: " . $i . "<br>";
    $i++;
} while ($i < 10);
```

Output

```
Il valore di i è: 0
Il valore di i è: 1
Il valore di i è: 2
Il valore di i è: 3
Il valore di i è: 4
Il valore di i è: 5
Il valore di i è: 6
Il valore di i è: 7
Il valore di i è: 8
Il valore di i è: 9
```

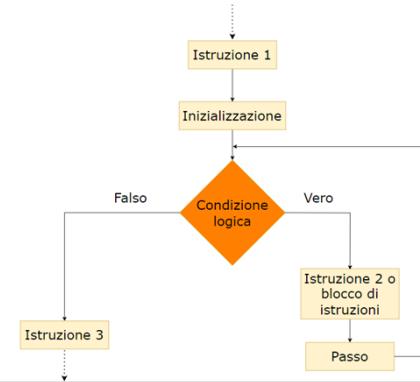
Prof. Montemurro

2iii Ciclo for

Sintassi generale del ciclo for

```
<?php
for (inizializzazione; condizione; passo) {
    // istruzioni eseguite se la condizione è vera
}
?>
```

Attenzione: la componente inizializzazione viene eseguita una sola volta.



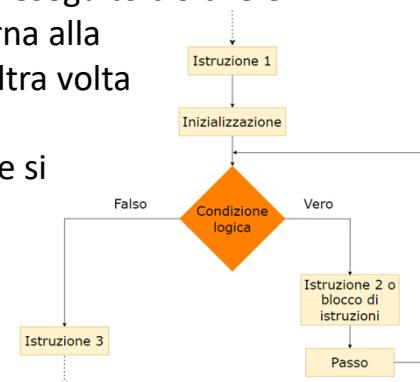
Prof. Montemurro

2iii Ciclo for

for: quando il programma arriva all'istruzione for, viene eseguito ciò che è specificato nella componente inizializzazione; in seguito viene valutata la condizione:

- se la condizione vale true, viene eseguito il blocco di istruzioni tra le parentesi graffe, ossia il **corpo del for**, e poi viene eseguito ciò che è specificato nella componente passo; dopo si ritorna alla condizione, e se vale di nuovo true, si ripete un'altra volta il corpo del for ed il passo;
- se la condizione vale false, si esce dal ciclo for e si esegue l'istruzione che segue il ciclo for.

Attenzione: la componente inizializzazione viene eseguita una sola volta.



Prof. Montemurro

2iii Ciclo for

Esempio

```
// Questo ciclo for stampa i numeri da 0 a 9
for ($i = 0; $i < 10; $i++) {
    echo "Il valore di i è: " . $i . "<br>";
}
```

Output

```
Il valore di i è: 0
Il valore di i è: 1
Il valore di i è: 2
Il valore di i è: 3
Il valore di i è: 4
Il valore di i è: 5
Il valore di i è: 6
Il valore di i è: 7
Il valore di i è: 8
Il valore di i è: 9
```

Prof. Montemurro

2iv Ciclo foreach

Ciclo foreach: si usa solo per gli array (*) e serve per scorrere i suoi **elementi** (ricordare che in generale l'elemento di un array è una **coppia chiave-valore**, cioè ogni valore ha un suo indice).

Esistono due sintassi per il ciclo foreach:

1. sintassi del ciclo foreach per ottenere solo i valori degli elementi dell'array;
2. sintassi del ciclo foreach per ottenere sia i valori che i relativi indici degli elementi dell'array.

(*) In realtà anche per gli oggetti di una classe che non vedremo (un oggetto è un'istanza di una classe).

Prof. Montemurro

2iv Ciclo foreach

Sintassi del ciclo foreach per ottenere solo i valori degli elementi dell'array

```
foreach ($nomeArray as $value) {
    // istruzioni
}
```

- Alla prima iterazione del ciclo foreach, il valore del primo elemento di \$nomeArray viene assegnato alla variabile \$value.
- Alla seconda iterazione del ciclo foreach, il valore del secondo elemento di \$nomeArray viene assegnato alla variabile \$value (quindi \$value non contiene più il **valore del primo elemento** di \$nomeArray in quanto **tale valore** è stato sovrascritto dal valore del secondo elemento di \$nomeArray).
- ...e così via per le altre iterazioni.

Prof. Montemurro

2iv Ciclo foreach

Esempio: ciclo foreach per ottenere solo i valori degli elementi dell'array

```
$voti = array(8.0, 6.5, 5.5, 7.0);
```

```
foreach ($voti as $value) {
    echo "$value <br>";
}
```

Output

8
6.5
5.5
7

Prof. Montemurro

2iv Ciclo foreach

Sintassi del ciclo `foreach` per ottenere sia i valori che i relativi indici degli elementi dell'array

```
foreach ($nomeArray as $key => $value) {
    // istruzioni
}
```

- Alla prima iterazione del ciclo `foreach`, il valore del primo elemento di `$nomeArray` viene assegnato alla variabile `$value` ed il suo indice viene assegnato alla variabile `$key`.
- ...e così via per le altre iterazioni.

Prof. Montemurro

2iv Ciclo foreach

Esempio: ciclo `foreach` per ottenere sia valori che i relativi indici degli elementi dell'array

```
$persona = array("cognome"=>"Rossi", "nome"=>"Toto",
    "nascita"=>2000, "professione"=>"impiegato");
```

```
foreach ($persona as $key => $value) {
    echo "$key: $value <br>";
}
```

Output

cognome: Rossi
nome: Toto
nascita: 2000
professione: impiegato

Prof. Montemurro

Form HTML: Ripasso GET e POST

<form **action**="URL" **method**="post">...</form>

action: attributo del tag `form` che indica dove si trova il programma PHP sul server che riceverà ed elaborerà i campi del form.

method: attributo del tag `form` che specifica (al browser) la modalità con cui inviare i campi del form al server. Esistono due metodi, get e post, le cui differenze sono:

GET	POST
Dati inviati <u>visibili</u> nell'URL	Dati inviati <u>non</u> visibili nell'URL
Pagina generata <u>salvabile</u> nei preferiti	Pagina generata <u>non</u> salvabile nei preferiti
Massima lunghezza URL: 3000 caratteri	URL di <u>qualsiasi</u> lunghezza

Nota bene: get non adatto per l'invio di dati sensibili (es. password ecc.).

Prof. Montemurro

Variabili superglobali

Variabili superglobali (o **variabili predefinite**, o **variabili d'ambiente**): variabili visibili in qualsiasi punto del codice PHP; queste variabili sono create direttamente dall'interprete PHP e sono raggruppate nei seguenti array associativi (`$_GET`, `$_POST`, `$_FILES`, `$_SERVER`, `$_COOKIE`, `$_SESSION`):

- **`$_GET`**: array associativo che contiene le variabili passate dal form HTML allo script PHP tramite il metodo get:
`<form action="URL" method="get">`
- **`$_POST`**: array associativo che contiene le variabili passate dal form HTML allo script PHP tramite il metodo post:
`<form action="URL" method="post">`

Prof. Montemurro

Variabili superglobali

Esempio 1: invio di dati dal form (lato client) allo script PHP (lato server) con metodo post

File HTML per inviare la coppia chiave-valore (qui il valore è il testo scritto dall'utente nella casella di testo)

```
<form action="cerca.php" method="post">
    Parola: <input type="text" name="parola" />
    <input type="submit" value="Cerca" />
</form>
```

File PHP per accedere al campo parola, cioè al valore memorizzato nella casella di testo il cui name è parola; tale valore è accessibile nello script PHP tramite la seguente variabile (ipotesi: utente scrive test e poi clicca su Cerca):

```
$valoreParola = $_POST["parola"];
echo $valoreParola; // test
```

Prof. Montemurro

Variabili superglobali

Esempio 2: invio di dati dal form (lato client) allo script PHP (lato server) con metodo get

File HTML per inviare la coppia chiave-valore (ipotesi: utente seleziona Visa)

```
<form action="invia.php" method="get">
    Scegli la carta di credito<br>
    <input type="radio" name="carta" value="visa"
           checked>Visa<br>
    <input type="radio" name="carta" value="mastercard">Mastercard<br>
    ...
    ...
```

Scegli la carta di credito
 Visa
 Mastercard

File PHP per accedere al valore inviato dal form allo script PHP, tale valore è accessibile nello script PHP tramite la seguente variabile:

```
$valoreCarta = $_GET["carta"];
echo $valoreCarta; // visa
```

Prof. Montemurro

Variabili superglobali

Esempio 3: solito get

File HTML

```
<form action="invia.php" method="get">
    Lingue conosciute<br>
    <input type="checkbox" name="cb1"
        value="ita">italiano<br>
    <input type="checkbox" name="cb2"
        value="ing">inglese<br>
    <input type="checkbox" name="cb3"
        value="fra">francese<br> ...
```

Lingue conosciute

italiano
 inglese
 francese

File PHP

```
echo $_GET["cb3"]."<br>"; // fra
//echo $_GET["cb2"]."<br>"; // errore se non c'è ing
echo $_GET["cb1"]; // ita
```

Prof. Montemurro

Variabili superglobali

Funzione predefinita `isset`: permette di verificare se l'utente ha assegnato un valore o meno ad una variabile definita nello script PHP.

Esempio

```
if (isset ($variabile)) {
    // istruzioni da eseguire se la variabile ha un
    // valore
} else {
    // istruzioni da eseguire se la variabile non ha un
    // valore
}
```

La funzione **isset** restituisce true se a \$variabile è stato assegnato un valore, mentre restituisce false se a \$variabile non è mai stato assegnato un valore, oppure ha se ha il valore null.

Prof. Montemurro

Variabili superglobali

...continuazione esempio 3: solito get

Per evitare errori tipo quello visto 2 slide fa conviene usare **isset**.

File HTML: il solito di prima

...

File PHP

```
if (isset($_GET["cb3"])) echo $_GET["cb3"] . "<br>";
if (isset($_GET["cb2"])) echo $_GET["cb2"] . "<br>";
if (isset($_GET["cb1"])) echo $_GET["cb1"] . "<br>";
```

Prof. Montemurro

Variabili superglobali

Approfondimento

- **\$_FILES**: array associativo che permette di gestire l'upload di file fatto dal form; si usa l'attributo **enctype** per specificare come devono essere codificati i dati del form quando si inviano al server, questo attributo si può usare solo col metodo post e può assumere solo i seguenti tre valori:
 1. **application/x-www-form-urlencoded** questo è il valore di default e si usa solo per i campi testuali, non per l'upload di file; tutti i caratteri vengono codificati prima dell'invio, (i) gli spazi vengono convertiti in simboli "+", e (ii) i caratteri speciali vengono convertiti in valori ASCII HEX;
 2. **multipart/form-data** questo valore è necessario quando si effettua l'upload di file tramite form;
 3. **text/plain** invia dati senza alcun codifica (non raccomandato).

Prof. Montemurro

Variabili superglobali

Approfondimento

- **`$_FILES`**

Le variabili contenute in `$_FILES` sono le seguenti sei:

1. `$_FILES['file']['name']` variabile che contiene il nome originale del file che l'utente ha scelto di caricare;
2. `$_FILES['file']['type']` variabile che contiene il tipo **MIME (Multipurpose Internet Mail Extensions)** del file;
Esempio: "image/gif"; questo tipo MIME non è tuttavia controllato sul lato PHP;
3. `$_FILES['file']['size']` variabile che contiene la dimensione in byte del file caricato;

Prof. Montemurro

Variabili superglobali

Approfondimento

- **`$_FILES`**

Le variabili contenute in `$_FILES` sono le seguenti sei:

4. `$_FILES['file']['tmp_name']` variabile che contiene il nome del file temporaneo in cui il file caricato è stato memorizzato sul server;
5. `$_FILES['file']['full_path']` variabile che contiene il percorso completo del file inviato dal browser (disponibile da PHP 8.1.0);
6. `$_FILES['file']['error']` variabile che contiene il codice di errore associato al caricamento del file.

Prof. Montemurro

Approfondimento

Variabili superglobali

- **`$_SERVER`**: array associativo che contiene le variabili passate allo script PHP dal server web. Alcune variabili contenute in **`$_SERVER`** sono le seguenti:
 1. **`$_SERVER['REMOTE_ADDR']`** che contiene l'indirizzo IP della macchina dell'utente da cui proviene la richiesta di esecuzione dello script PHP;
 2. **`$_SERVER['SERVER_SOFTWARE']`** che contiene una stringa con il nome del server web che esegue lo script PHP (es. Apache/2.2.24);
 3. **`$_SERVER['HTTP_USER_AGENT']`** che contiene una stringa con il nome del browser utilizzato dall'utente;
 4. **`$_SERVER['SERVER_NAME']`** che contiene una stringa con il nome della macchina su cui gira il server web;
 5. **`$_SERVER['PHP_SELF']`** che contiene una stringa con il nome dello script PHP in esecuzione.

Prof. Montemurro

Approfondimento

Variabili superglobali

- **`$_SERVER`**: array associativo che contiene variabili che possono essere usate:
 1. per registrare gli accessi al server attraverso i **file di log** (log = registro);
 2. per produrre successivamente le statistiche relative ai visitatori (es. da dove si collegano e quale browser usano).

Gli array associativi predefiniti del linguaggio PHP devono essere scritti con i caratteri tutti maiuscoli. Gli indici delle variabili, tra parentesi quadre, sono racchiusi da apici singoli o doppi:

Esempio: ottenere indirizzo IP dell'utente

```
<?php echo $_SERVER['REMOTE_ADDR'] . "<br>" ?>
```

Prof. Montemurro

Variabili superglobali

Approfondimento

- **\$_COOKIE**: array associativo che contiene le variabili passate allo script PHP tramite i cookie.

Cookie: file di testo di piccole dimensioni, memorizzati sul computer dell'utente (client) su richiesta esplicita del server, questi file vengono inviati al server quando l'utente visita di nuovo il sito web. I cookie contengono alcune informazioni (es. informazioni sulle pagine web visitate sul sito sul quale si sta navigando; informazioni per riconoscere univocamente il client; informazioni sulle preferenze dell'utente quali tema chiaro/scuro, voci memorizzate nel carrello di un sito di e-commerce, username e password ecc.)

Prof. Montemurro

Variabili superglobali

Approfondimento

- **\$_SESSION**: array associativo che contiene le variabili utilizzate per implementare il concetto di sessione.
- Sessione** (o **sessione di lavoro**): svolgimento, in un arco temporale, di un insieme di attività tra loro correlate, e tali che una o alcune di queste che sono preliminari ad altre (es. un qualunque servizio di accesso a informazioni richiede preliminarmente l'identificazione dell'utente; fatto ciò, l'utente può accedere al servizio per accedere ai dati sfruttando gli script PHP sul server).
- Nota bene**: le variabili di uno script, di norma, hanno validità limitata al solo script PHP nel quale sono state create; terminata l'esecuzione dello script PHP, esse vengono distrutte. A questa regola fanno eccezione le variabili dell'array **\$_SESSION** la cui validità può essere estesa a diversi script PHP. Questa caratteristica è sfruttata per realizzare sessioni di lavoro.

Prof. Montemurro

Classi e Oggetti

aa

Prof. Montemurro

Paradigma Imperativo di Programmazione Orientata agli Oggetti

Paradigma orientato agli oggetti: analizza il problema cercando di individuare gli **oggetti** che lo compongono e le associazioni tra essi. Tali oggetti possono essere oggetti reali (es. automobile) oppure concetti (o oggetti astratti) (es. figura geometrica, operazione matematica). Più oggetti comunicano tra di loro scambiandosi dei **messaggi**.

Esempi: linguaggio di programmazione C++, Java, PHP.

Prof. Montemurro

Struttura di un Oggetto

La **struttura di un oggetto** è composta:

1. dagli **attributi** i quali sono elementi che descrivono le caratteristiche dell'oggetto;
2. dai **metodi** i quali sono i comportamenti e le funzionalità che l'oggetto mette a disposizione.



Prof. Montemurro

Paradigma Imperativo di Programmazione Orientata agli Oggetti

Esempio di oggetto: aereo

- i. Attributi: modello, numero di posti, tipo di carburante, velocità massima ecc.
- ii. Metodi: decolla, atterra, rifornisci ecc.



Prof. Montemurro

Accesso ai DB con MySQLi

Obiettivo: eseguire la connessione al DB ed eseguire le operazioni di interrogazione e manipolazione su un DB con l'**estensione MySQLi** (**SQL** sta per **Structured Query Language**).

Connessione al server: la **classe mysqli** rappresenta la connessione tra il PHP ed il **DB MySQL**. La prima operazione eseguita dallo script PHP è la connessione al **server MySQL**. Si definisce l'oggetto **\$conn** (o istanza della classe **mysqli**) si ottiene con l'operatore **new**:

```
$conn = new mysqli($host, $username, $password,
                   $db_nome);
```

Prof. Montemurro

Accesso ai DB con MySQLi

```
$conn = new mysqli($host, $username, $password,
                   $db_nome);
```

Gli argomenti dell'istanza della classe **mysqli** sono i seguenti quattro:

1. **\$host** rappresenta l'indirizzo IP o il nome del server su cui è in esecuzione il server MySQL (se si utilizza un server di sviluppo su computer locale, si indica **localhost**);

Prof. Montemurro



Prof. Montemurro

Sistemi Informativi

Classe 5ASI
ITCG Fermi

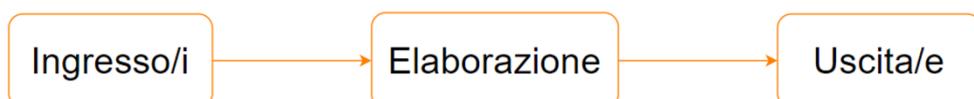
Prof. Montemurro

Premessa: Dati e la Loro Elaborazione

Dato: descrive aspetti elementari di entità o fenomeni; è la misura di un evento, un fatto oggettivo espresso con numeri o parole.

Informazione: insieme di dati elaborati. Tutto ciò che possiede un significato per l'uomo, e che viene conservato e/o comunicato in vista di un'utilità pratica immediata o futura.

Elaborazione: trattamento dei dati per ottenere le informazioni (ess. trascrizione dei dati su carta; calcolo matematico ecc.).

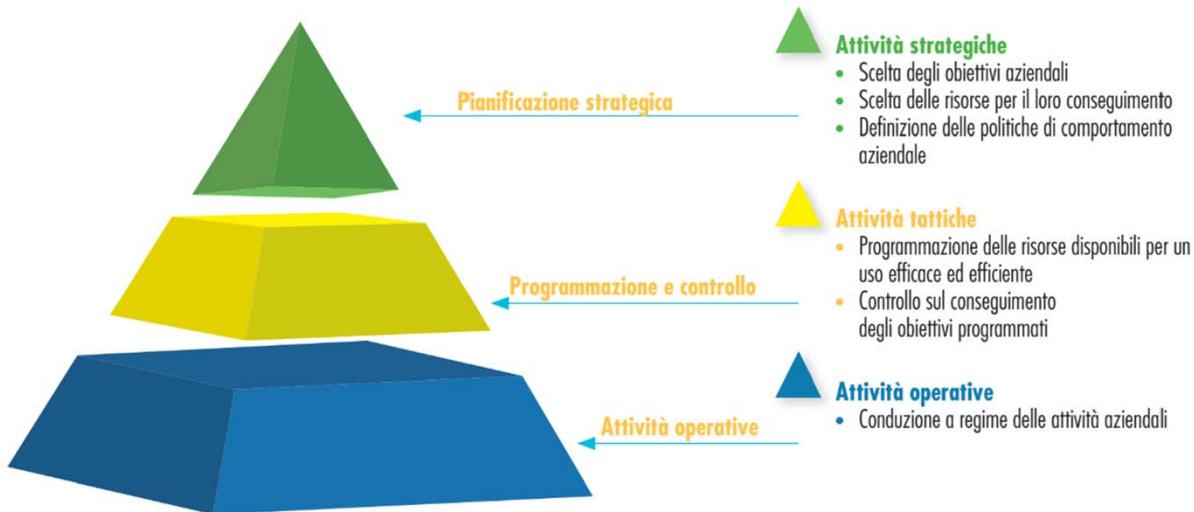


Ingressi: dati.

Uscite: informazioni; se elaborate ulteriormente esse sono dati di ingresso delle elaborazioni che saranno fatte successivamente.

Prof. Montemurro

Attività Aziendali: Piramide di Anthony



Prof. Montemurro

Attività Aziendali: Piramide di Anthony

- Attività strategiche:** consistono nella pianificazione strategica ossia nel processo di decisione sulla base degli obiettivi di medio-lungo periodo, sull'acquisizione delle risorse per conseguirli, sulle politiche da adottare nell'uso di tali risorse. La pianificazione strategica richiede attitudine all'analisi ma anche doti di intuito, è diretta verso l'esterno e difficilmente strutturabile.
- Attività tattiche:** sono legate all'amministrazione corrente dell'azienda, sono le attività che possiamo definire di **programmazione e controllo** perchè stabiliscono l'assegnazione delle risorse, e controllano il loro corretto uso per il conseguimento degli obiettivi programmati. Sono attività per lo più ripetitive e sistematiche, orientate verso l'interno dell'azienda.

Prof. Montemurro

Attività Aziendali: Piramide di Anthony

3. **Attività operative:** consistono nello svolgimento delle operazioni correnti, cioè nell'esecuzione delle attività che consentono all'azienda di funzionare. Sono attività molto ripetitive, in cui il margine di discrezionalità decisionale è molto ridotto, e sono molto legate al tempo.

Prof. Montemurro

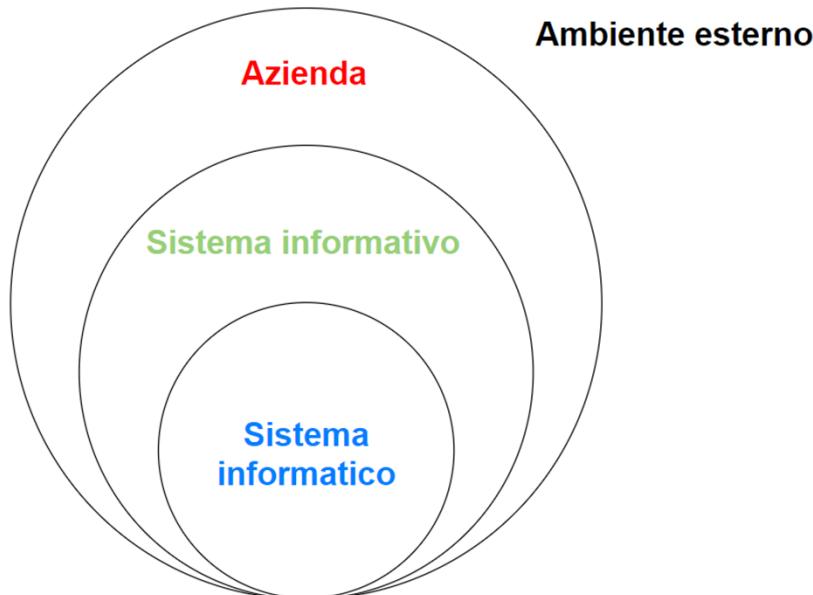
Definizione di Flussi Informativi

Flussi informativi: sono i flussi, gli scambi delle informazioni che avvengono all'interno dell'azienda, ed in particolare nella piramide di Anthony. Ce ne sono di due tipi:

1. **flussi informativi orizzontali**, cioè scambi di informazioni tra persone che appartengono allo stesso livello della piramide;
2. **flussi informativi verticali**, cioè scambi di informazioni tra persone che appartengono a livelli diversi della piramide. In genere questi flussi riguardano le eccezioni; in particolare:
 - i. nei flussi informativi dal basso verso l'alto vengono trasmesse le informazioni per segnalare le eccezioni (es. merce inattesa in magazzino);
 - ii. nei flussi informativi dall'alto verso il basso vengono trasmesse le informazioni per gestire tali eccezioni (es. merce mandata indietro).

Prof. Montemurro

Relazione Gerarchica tra...



Prof. Montemurro

Definizione di SIA

Sistema informativo aziendale SIA: insieme

1. di **risorse** umane e materiali (es. impianti, materie prime);
2. di **principi e regole** (es. norme e linee guida per il funzionamento del SIA);
3. delle **procedure** per:
 - i. la **raccolta** (o **acquisizione**) dei dati,
 - ii. l'**archiviazione** dei dati e delle informazioni,
 - iii. l'**elaborazione** dei dati al fine di produrre informazioni,
 - iv. lo **scambio** (o **trasmissione**) dei dati e delle informazioni,
 - v. la **presentazione** dei dati e delle informazioni.

Queste **informazioni** sono necessarie alle **attività operative, di gestione, di programmazione, di controllo, e di valutazione**.

Dunque il SIA non si limita agli aspetti tecnologici, ma include anche elementi umani, organizzativi e culturali.

Prof. Montemurro

Definizione di Sistema Informatico

Sistema informatico: è la parte informatizzata del SIA costituita dall'insieme delle **applicazioni software** e degli **strumenti hardware** che gestiscono i dati e i flussi informativi. Tale gestione consiste:

1. nella **raccolta** (o **acquisizione**) dei dati,
 2. nell'**archiviazione** dei dati e delle informazioni,
 3. nell'**elaborazione** dei dati al fine di produrre informazioni,
 4. nello **scambio** (o **trasmissione**) dei dati e delle informazioni,
 5. nella **presentazione** dei dati e delle informazioni,
- mediante le tecnologie informatiche e della comunicazione (ICT).

Prof. Montemurro

Fasi del Ciclo di Vita di un SIA

1. Studio di fattibilità
2. Progetto esecutivo
3. Realizzazione
4. Test o collaudo
5. Avviamento
6. Esercizio

Il SIA segue l'azienda per tutta la sua vita; il suo ciclo di vita può durare anche alcuni anni.

Prof. Montemurro

1 Studio di Fattibilità

Gestione: responsabile di progetto + gruppo di sviluppo informatico + cliente

Obiettivi

- Definizione dei **costi** e dei **tempi**, dei **rischi** delle diverse alternative possibili
- Stabilire la **priorità** di realizzazione delle varie componenti del SIA
- Definizione del contratto col cliente

Domande guida

- **Cosa** deve fare il SIA?
- **Cosa** si aspetta il cliente?
- Quante e quali persone saranno coinvolte?
- Quante e quali processi saranno coinvolte?

Fase finale: firma del contratto da parte del cliente e dell'esecutore del lavoro

Prof. Montemurro

2 Progetto Esecutivo

Gestione: responsabile di progetto + gruppo di sviluppo informatico

Obiettivi

- Progettazione del DB (modello concettuale dei dati, modello logico dei dati)
- Progettazione delle applicazioni, cioè le caratteristiche dei programmi applicativi (es. interfaccia utente)
- Redazione della documentazione tecnica in cui si descrive l'architettura del SIA

Domanda guida: il SIA **come** deve operare?

Prof. Montemurro

3 Realizzazione

Gestione: responsabile di progetto + gruppo di sviluppo informatico

Obiettivi

- Acquisizione delle piattaforme hardware e di comunicazione
- Sviluppo e installazione delle applicazioni software

4 Test e Collaudo

Gestione: responsabile di progetto + responsabili del collaudo.

Obiettivo: verificare, in ogni possibile condizione operativa che può presentarsi, il corretto funzionamento di tutte le componenti del SIA.

Prof. Montemurro

5 Avviamento

Gestione: responsabile di progetto + gruppo di sviluppo informatico

Obiettivi

- Eseguire attività necessarie per rendere operativo il SIA
- Garantire affidabilità ed integrità dei dati dopo aver formato il personale e dopo aver eseguito la migrazione (o spostamento) delle procedure informatiche dal vecchio SIA al nuovo SIA

Prof. Montemurro

6 Esercizio

Obiettivo

- Fare manutenzione; i tre tipi di manutenzione del software sono:
 1. la **manutenzione correttiva** la quale riguarda la correzione di eventuali errori non riscontrati in fase di realizzazione;
 2. la **manutenzione evolutiva** la quale riguarda la modifica del software per tener conto di eventuali cambiamenti nelle esigenze degli utenti finali e nella normativa;
 3. la **manutenzione adattativa** la quale riguarda la modifica del software per adeguarlo alle innovazioni tecnologiche.

Prof. Montemurro

Svantaggi di un SIA Tradizionale

1. Non unicità del DB per cui si ha **ridondanza dei dati**, ossia gli stessi dati sono memorizzati più volte perché non c'è un unico DB, ma ci sono diversi archivi (o **file**), uno per ciascun applicativo. Ogni funzione interna all'azienda ha il suo applicativo (es. manifattura, logistica, finanza, risorse umane ecc.).

File: archivio memorizzato su una memoria di massa.

Prof. Montemurro

Svantaggi di un SIA Tradizionale

1. Non unicità del DB...

Problemi della ridondanza dei dati:

- i. **costi di memorizzazione** visto che ci sono più copie degli stessi dati; ogni variazione di un dato duplicato implica l'aggiornamento di tutti i file ove il dato è presente (operazione complessa e costosa: devo pagare i dipendenti per fare lo stesso lavoro più volte);
- ii. **rischio d'incongruenza dei dati:** dati che fanno riferimento allo stesso oggetto, ma che non sono aggiornati (es. età = 48 anni su un archivio, e 49 su un altro);
- iii. l'incongruenza dei dati comporta l'**inconsistenza dei dati**, cioè non si è più in grado di riconoscere quali, tra i dati incongruenti, siano quelli corretti e quali quelli sbagliati.

Prof. Montemurro

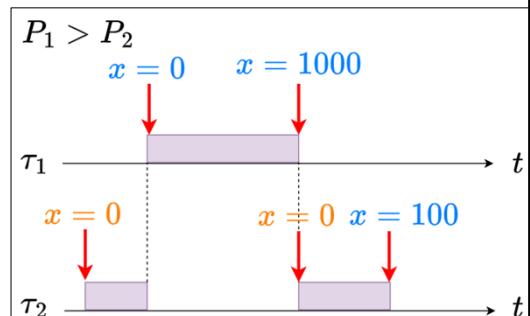
Svantaggi di un SIA Tradizionale

1. Non unicità del DB...

Problema dell'accesso concorrente ai dati: si parla di **concorrenza** quando due o più programmi accedono allo stesso dato in un file contemporaneamente; quando ciò accade si parla di **accesso concorrente** (o **accesso senza mutua esclusione**) al dato, e si manifesta il seguente problema: quando più programmi modificano lo stesso dato, viene salvata solo l'ultima modifica per cui si perde un'operazione.

Esempio: task 1 mette € 1000, task 2 € 100

Sia x il saldo di un CC; il task 1 ha priorità più alta del task 2, cioè $P_1 > P_2$, per cui, se il task 2 è in esecuzione, quando parte il task 1, questo blocca l'esecuzione del task 2.



Prof. Montemurro

Svantaggi di un SIA Tradizionale

2. Non pacchettizzato, per cui non personalizzabile e poco flessibile.
Prodotto pacchettizzato: prodotto progettato per essere usato da molte aziende e persone aventi bisogni diversi tra loro.
3. Non modulare.

Prof. Montemurro

Definizione di SIA Integrato (o ERP)

Sistema informativo aziendale integrato (o **Enterprise Resource Planning** **ERP**): insieme

1. di **risorse** umane e materiali (es. impianti, materie prime);
2. di **principi e regole** (es. norme e linee guida per il funzionamento del SIA);
3. delle **procedure** per:
 - i. la **raccolta** (o **acquisizione**) dei dati,
 - ii. l'**archiviazione** dei dati e delle informazioni,
 - iii. l'**elaborazione** dei dati al fine di produrre informazioni,
 - iv. lo **scambio** (o **trasmissione**) dei dati e delle informazioni,
 - v. la **presentazione** dei dati e delle informazioni.

Queste **informazioni** sono necessarie alle **attività operative, di gestione, di programmazione, di controllo, e di valutazione**.

Prof. Montemurro

Definizione di SIA Integrato (o ERP)

Caratteristiche dell'ERP (che il SIA tradizionale non ha)

1. **DB unico** (distribuito o no) per cui si evita la ridondanza dei dati, l'incongruenza dei dati, l'inconsistenza dei dati, e si garantisce l'accesso mutuamente esclusivo ad un dato (un dato è accessibile solo ad un task alla volta).
2. **Prodotto pacchettizzato**, cioè progettato per essere usato da molte aziende e persone aventi bisogni diversi tra loro, per cui è personalizzabile e flessibile, in termini di procedure, in base ai bisogni specifici di ogni azienda.
3. **Prodotto modulare**, cioè composto da moduli funzionali, i principali sono: **contabilità, controllo della gestione, gestione del personale, gestione degli acquisti, gestione dei magazzini, gestione dei progetti, gestione delle vendite (SFA), gestione delle relazioni con i fornitori (SCM), gestione delle relazioni coi clienti (CRM), gestione degli asset**.

Un'azienda può acquistare anche solo alcuni di questi moduli.

Prof. Montemurro

DB Accentrati/Distribuiti

Database accentrato: è memorizzato su un unico computer.

Database distribuito: è memorizzato su memorie di massa di diversi computer anche fisicamente lontani, però, a livello logico, è un tutt'uno.

Prof. Montemurro



Prof. Montemurro

Fondamenti di Networking e Reti Informatiche Aziendali

Classe 5ASI
ITCG Fermi

Prof. Montemurro

Definizione di Rete Informatica e Networking

Rete informatica: insieme di **hardware**, **software**, e **cablaggio** i quali permettono a più dispositivi di elaborazione di comunicare tra loro.

I computer collegati in rete possono (1) scambiarsi file, e (2) condividere periferiche (es. stampanti, dispositivi di archiviazione collegati alla rete NAS ecc.).

Networking: connessioni tra diverse reti informatiche.

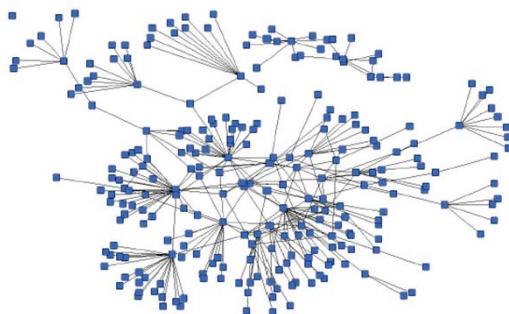
Prof. Montemurro

Definizione di Rete Informatica e Networking

Una rete (anche non informatica) è costituita da:

1. **nodi** i quali sono le unità che costituiscono la rete. Nel caso di una rete informatica, un nodo è una qualsiasi entità che genera/smista/riceve informazioni sui canali di comunicazione della rete stessa (esempi di nodi: PC, tablet, smartphone, stampanti, switch, router ecc.);
2. **connessioni** tra i nodi.

- Una connessione collega una coppia di nodi.
- Ogni nodo può avere più connessioni.



Prof. Montemurro

Classificazione delle Reti in Base all'Estensione Geografica

Rete informatica	Distanza coperta (m)
LAN (Local Area Network (o rete locale))	500
CAN (Campus Area Network)	1000
MAN (Metropolitan Area Network)	Area metropolitana
WAN (Wide Area Network (o rete geografica))	Area transcontinentale

Esempio di LAN: rete informatica in un'unica sede aziendale.

Esempio di WAN: due o più reti informatiche LAN, collegate tra loro, che si trovano in sedi differenti nel mondo.

Prof. Montemurro

Componenti di Base di una Rete Aziendale

1. Almeno due host.

Host è un qualunque dispositivo dotato di CPU in grado di connettersi ad una rete (es. computer, tablet, smartphone, console per videogiochi ecc.).

2. Almeno una scheda di rete su ciascun host.

Scheda di rete: dispositivo che permette di connettere un host ad una rete.

3. Un **mezzo di collegamento:** cavi oppure connessione senza cavo (wireless).

4. Un **software di rete.**

5. Un **punto di aggregazione**, cioè un dispositivo in grado di connettere tutti i cavi; oggi i dispositivi che fanno da punto di aggregazione sono gli **switch**.

Prof. Montemurro

Modelli di Reti Informatiche Aziendali

Il sistema informatico di un sistema informativo aziendale può basarsi sui seguenti modelli di reti informatiche:

1. modello mainframe-terminal;
2. modello client-server;
3. modello peer-to-peer (P2P);
4. modelli ibridi che prevedono l'uso sia del modello client-server che P2P.

Prof. Montemurro

Modelli di Reti Informatiche Aziendali

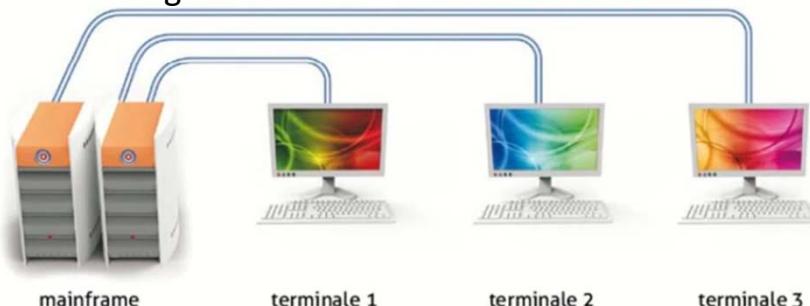
Modello mainframe-terminale (di una rete informatica aziendale): questo modello prevede la presenza di:

1. una macchina chiamata **mainframe** la quale è in grado di eseguire operazioni di elaborazione e memorizzazione dei dati che poi verranno distribuiti all'utente;
2. **terminali** i quali non sono in grado di elaborare informazioni, sono usati solo visualizzare ed inserire dati dagli utenti.

Svantaggi

- i. hardware complesso e costoso;
- ii. sistema poco flessibile.

Prof. Montemurro

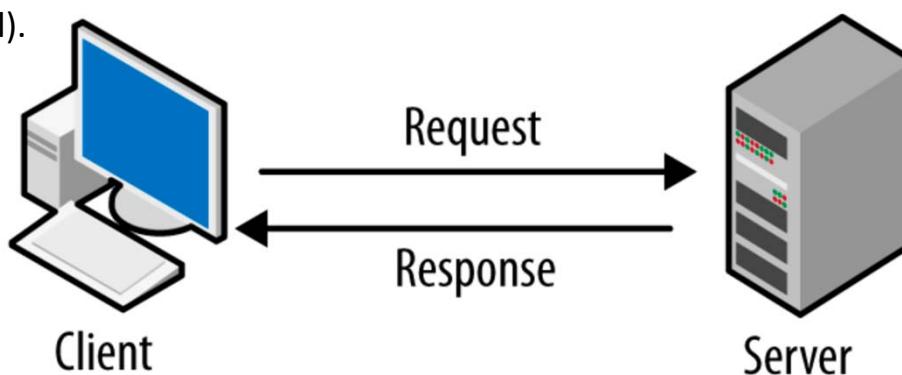


mainframe terminale 1 terminale 2 terminale 3

Modelli di Reti Informatiche Aziendali

Modello client-server (di una rete informatica aziendale): architettura software costituita da due applicazioni:

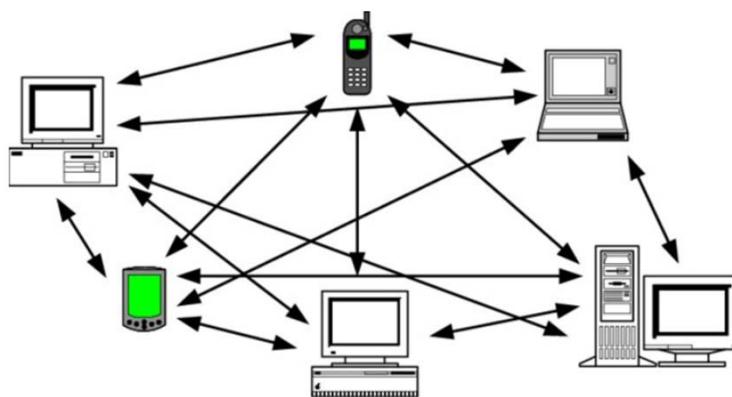
1. **client**: applicazione che richiede un servizio (es. Gmail sul browser del nostro PC);
2. **server**: applicazione che fornisce un servizio (es. server Google fornisce Gmail).



Prof. Montemurro

Modelli di Reti Informatiche Aziendali

Modello peer-to-peer (P2P) (di una rete informatica aziendale): il client ed il server risiedono entrambi su ciascun computer della rete informatica.



Prof. Montemurro

Cablaggio di Reti Informatiche Aziendali

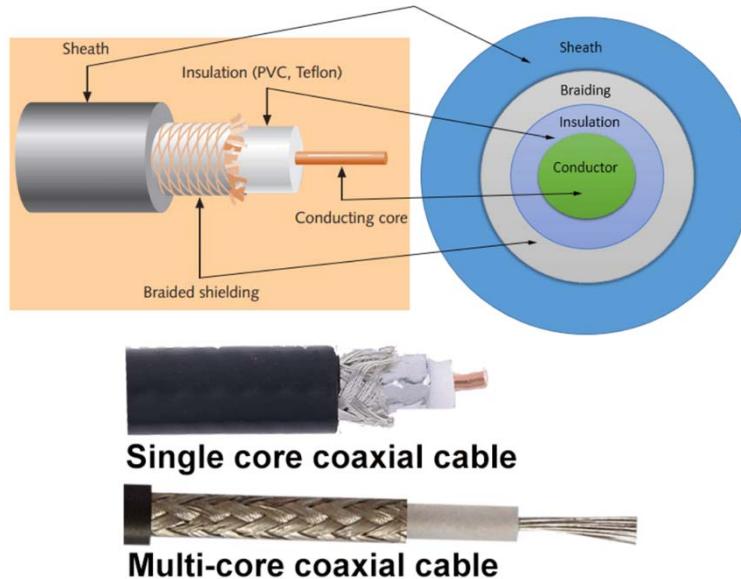
Cablaggio: modalità con le quali si effettuano i collegamenti fisici tra i dispositivi di rete.

Tipi di cavi

1. **Cavo coassiale** il quale è poco utilizzato
2. **Doppino in rame** (o **cavo Ethernet**) il quale si usa nelle reti LAN, il suo spinotto terminatore si chiama **RJ45**, è conforme a vari standard di rete (**UTP**, **FTP**, **STP**; questi si differenziano per la schermatura dalle interferenze elettromagnetiche EM), e la distanza massima coperta non supera i 100 metri
3. **Fibra ottica** la quale si usa per le linee ad alta velocità sulle quali vengono scambiate grandi quantità di dati, e la distanza massima coperta è di circa 10 chilometri

Prof. Montemurro

Cablaggio: Cavi Coassiali di Rete



Prof. Montemurro

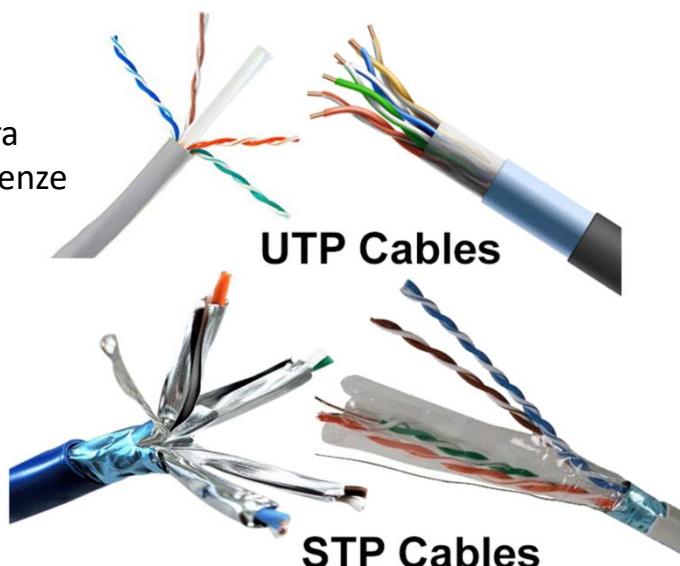
Cablaggio: Doppino in Rame (o Cavo Ethernet) di Rete

1. Cavi Ethernet UTP (Unshielded Twisted Pair)

Twisted Pair): tutte le coppie di cavi sono avvolte in un'unica guaina di plastica (no schermatura metallica protettiva dalle interferenze elettromagnetiche).

2. Cavi Ethernet STP (Shielded Twisted Pair)

Twisted Pair): ogni coppia è avvolta da un ulteriore strato di protezione metallica, poi tutte le coppie sono avvolte in un'unica guaina esterna metallica e da una di plastica (schermatura massima dalle interferenze EM).



Prof. Montemurro

Cablaggio: Doppino in Rame (o Cavo Ethernet) di Rete

3. **Cavi Ethernet FPT (Foiled Twisted Pair)**: tutte le coppie sono avvolte in un'unica guaina esterna metallica e da una di plastica, le singole coppie non sono schermate (schermatura intermedia dalle interferenze EM rispetto ai cavi UTP e STP).

Prof. Montemurro

Cablaggio: Spinotto Terminatore RJ45 del Doppino in Rame



Fig. 2. RJ-45 connector

Prof. Montemurro

Cablaggio: Fibra Ottica



Prof. Montemurro

Dispositivi di Rete

Dispositivi di rete: dispositivi che permettono la comunicazione tra gli host.

Tipi di dispositivi di rete

1. Schede di rete
2. Hub
3. Switch
4. Router

Prof. Montemurro

Dispositivi di Rete: Schede di Rete

Scheda di rete: dispositivo che permette di connettere un host ad una rete; in genere sono già montate sulla scheda madre, ed hanno una **porta RJ45** dove viene connesso il cavo Ethernet.

Tipi di schede di rete

1. **Schede di rete Ethernet** le quali sono obsolete; velocità massima di comunicazione pari a 10 Mbps
2. **Schede di rete Fast Ethernet** le quali permettono una velocità massima di comunicazione pari a 100 Mbps
3. **Schede di rete Gigabit Ethernet** le quali permettono una velocità massima di comunicazione pari a 1000 Mbps (1 Gbps)
4. **Schede di rete senza fili** (o **wireless**, o **Wi-Fi**)

Prof. Montemurro

Dispositivi di Rete: Schede di Rete con Porta RJ45



Prof. Montemurro

Dispositivi di Rete: Schede di Rete Wi-Fi



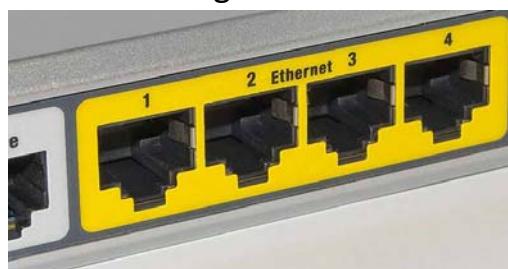
Prof. Montemurro

Dispositivi di Rete: Hub e Switch

Hub e switch: sono entrambi dispositivi elettronici in grado di connettere tutti i cavi della rete per cui permettono la comunicazione tra gli host; ciascun cavo di rete è collegato ad un host diverso.

Porta fisica: è una connessione fisica (es. porta RJ45), un'interfaccia fisica dove si può inserire un cavo di rete che in genere è il cavo Ethernet.

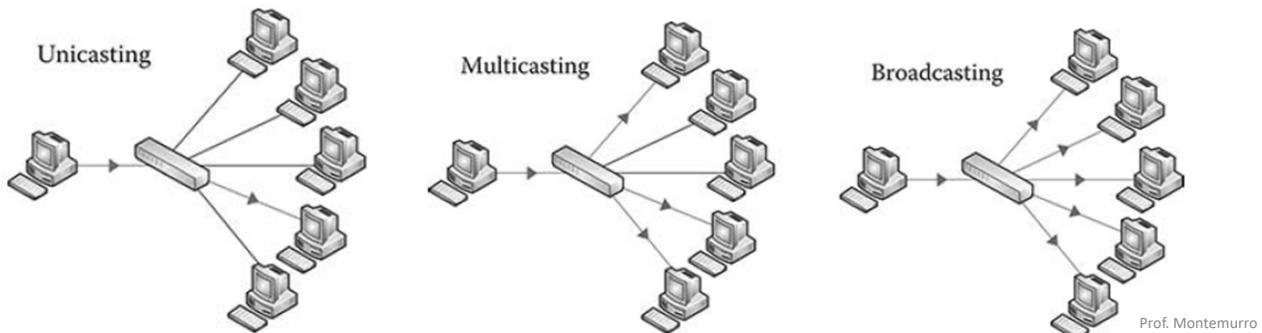
Porte RJ45



Prof. Montemurro

Digressione: Metodi di Distribuzione dei Dati

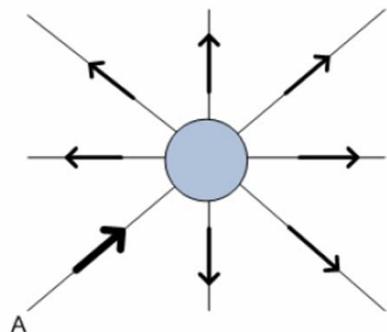
1. **Unicasting:** trasmettere dati tra una sorgente ed un destinatario (es. mando un'email ad un singolo destinatario)
2. **Multicasting:** trasmettere dati contemporaneamente ad un gruppo selezionato di destinatari (es. invio email ad un paio di amici)
3. **Broadcasting:** trasmettere dati contemporaneamente a tutti i nodi della rete (es. radio, televisione)



Dispositivi di Rete: Hub

Hub: dispositivo elettronico che riceve un messaggio su una porta e lo invia su tutte le porte, tranne alla porta sulla quale il messaggio è stato ricevuto (**operazione di flooding**). La distribuzione dei dati è di tipo broadcast.

Esempio: l'hub in figura riceve un messaggio sulla porta A, e lo invia su tutte le altre porte.



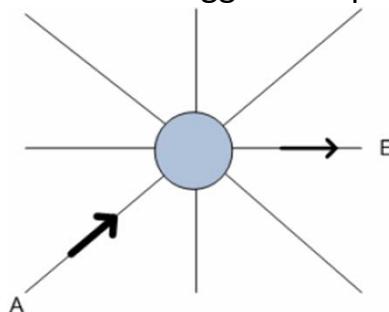
Prof. Montemurro

Dispositivi di Rete: Switch



Switch (o commutatore): dispositivo elettronico che riceve un messaggio su una porta e lo invia solo su una sola porta che viene specificata all'interno del messaggio stesso. La distribuzione dei dati è di tipo unicast.

Esempio: l'hub in figura riceve un messaggio sulla porta A, e lo invia solo sulla porta E.



Prof. Montemurro

Dispositivi di Rete: Switch



La principale caratteristica di uno switch è la sua **banda di trasmissione**, cioè quanti megabit al secondo (Mbps) riesce a trasmettere.

Esempio: switch con banda di trasmissione di 10 Mbps significa che ciascun utente potrà inviare/ricevere 10 megabit al secondo.

Patch panel



Switch

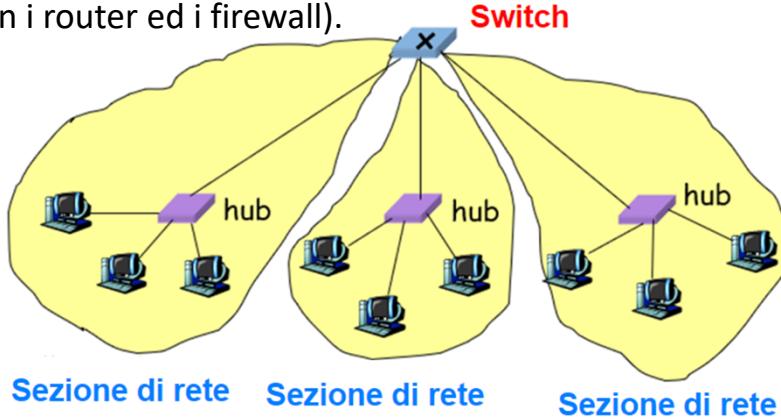
Prof. Montemurro

Dispositivi di Rete: Switch



Sezione di rete (o segmento di rete): insieme di dispositivi di rete che condividono lo stesso cavo di rete (se la rete è una rete cablata).

Gli switch consentono di creare e connettere tra loro più sezioni di rete (si può fare anche con i router ed i firewall).



Prof. Montemurro

Dispositivi di Rete: Router



Router: dispositivo elettronico che smista, cioè instrada i dati in genere verso l'esterno di una rete locale (LAN).

Esempio: i router sono usati per collegare due LAN tra loro oppure per collegare una LAN a internet.

Dunque i router sono usati soprattutto per collegare reti fisicamente distanti, cioè reti geografiche (WAN).

Ci sono diversi percorsi che i dati possono prendere, il router sceglie i percorsi più efficaci, cioè quelli in cui si ha il minor spreco di risorse (es. tempo). Se cade la connessione, per non interrompere il traffico, un router sorgente può definire anche un eventuale percorso alternativo.

Prof. Montemurro

Topologie di Rete

Topologia di rete: descrive (1) posizione (disposizione geometrica) di tutti i nodi che fanno parte della rete, (2) tutti i collegamenti fisici da realizzare per connettere i nodi.

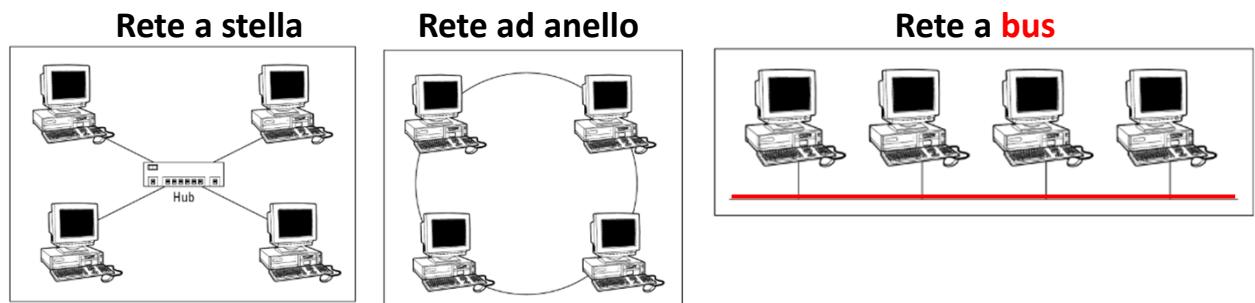
Parametri topologici di una rete

1. **Numero di nodi**
2. **Numero di canali trasmissivi**
3. **Ridondanza delle connessioni;** maggiore è il numero di connessioni alternative per raggiungere una stessa destinazione, maggiore è la **tolleranza ai guasti**. Infatti, se una connessione si rompe, ce n'è un'altra alternativa.

Prof. Montemurro

Tipi di Topologie di Rete

1. **Rete a stella**
2. **Rete ad anello**
3. **Rete a bus**

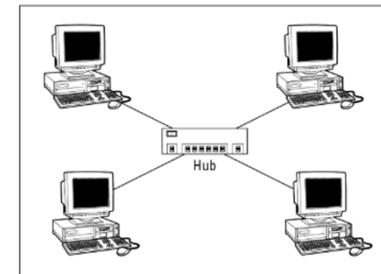


Prof. Montemurro

Topologie di Rete: Rete a Stella

Rete a stella: il numero di canali è uguale al numero di nodi meno uno ($n = c - 1$). Al centro della stella si trova un dispositivo concentratore (switch o hub).

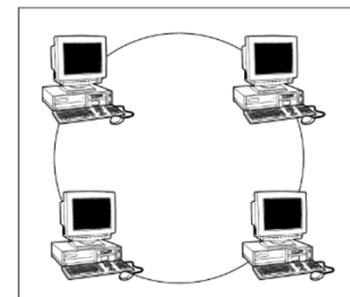
Svantaggio: tolleranza ai guasti inesistente; nel caso in cui il canale si guasti, la funzionalità della rete viene compromessa.



Prof. Montemurro

Topologie di Rete: Rete ad Anello

Rete ad anello: ogni nodo è collegato con altri nodi, in modo da formare una struttura circolare. Ci sono due percorsi alternativi per raggiungere una stessa destinazione. Ogni scambio di informazioni coinvolge anche nodi della rete che non sono direttamente interessati ai messaggi, nonostante ciò devono cooperare per permettere la trasmissione dei messaggi.

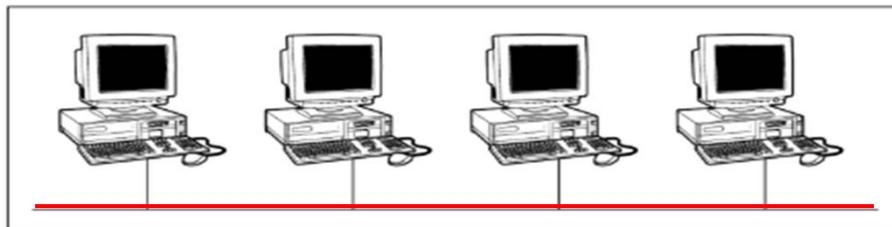


Prof. Montemurro

Topologie di Rete: Rete a Bus

Rete a bus: i computer sono connessi a un unico canale comune, condiviso da tutte le comunicazioni. Un solo nodo alla volta può trasmettere sul canale. Tutti i nodi possono valutare i messaggi, ma solo il nodo che riconosce di essere il destinatario acquisisce il messaggio.

Bus: insieme di fili ordinati che permettono la trasmissione di dati.



Prof. Montemurro

Collisioni di Rete

Collisione di rete: quando più dispositivi condividono lo stesso mezzo trasmissivo (es. cavo di rete o mezzo wireless), può accadere che due o più dispositivi tentino di inviare un messaggio contemporaneamente. Poiché su un mezzo trasmissivo può viaggiare un messaggio per volta, quando ne arriva più di uno, tali messaggi vengono distrutti o vengono corrotti (cioè non sono più uguali ai messaggi originali) per cui vengono scartati quando raggiungono i rispettivi destinatari. Questo conflitto tra più dispositivi che si verifica quando tentano di inviare contemporaneamente un messaggio sul mezzo trasmissivo in comune si chiama collisione di rete.

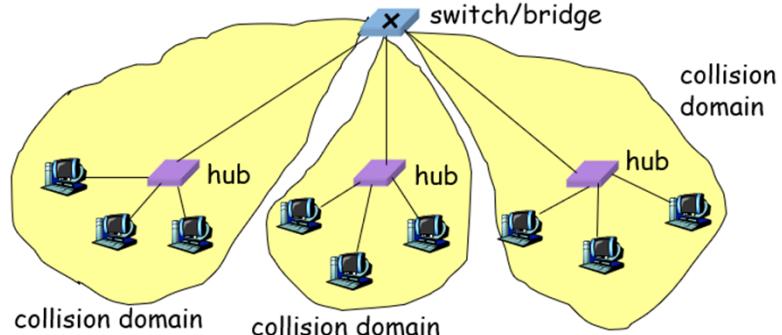
Esempio: due amici usano i walkie-talkies; se tentano di comunicare contemporaneamente, quello che si sente è rumore. Questa è una collisione di rete.

Prof. Montemurro

Dominio di Collisione

Dominio di collisione: insieme dei dispositivi che condividono lo stesso mezzo trasmissivo per cui, in fase di trasmissione del messaggio, si potrebbero verificare collisioni di rete.

Gli switch servono per creare domini di collisione; così facendo previene le eventuali collisioni tra i diversi domini di collisione grazie alla **bufferizzazione** (memorizzazione temporanea dei messaggi provenienti contemporaneamente da più dispositivi nella **memoria interna dello switch**; ciò permette di inviare un messaggio alla volta sul mezzo trasmissivo anche se più messaggi arrivano contemporaneamente).



Prof. Montemurro

Larghezza di Banda

Larghezza di banda: quantità di dati che possono circolare in un dato periodo di tempo, in un mezzo trasmissivo; si misura in bit/s (o bps, o bitrate). La larghezza di banda dipende anche dal tipo di mezzo trasmissivo (es. cavo Ethernet, fibra ottica, wireless).

Congestione della rete: eccesso di traffico che blocca i percorsi dei dati lungo la rete.

Quando viene utilizzato più del 50% delle larghezza di banda, il numero di collisioni tende ad aumentare rapidamente causando congestioni (il tempo per stampare i file si allunga, le applicazioni si aprono lentamente e gli utenti devono aspettare). Quando questo valore raggiunge e supera il 60%, la rete rallenta drasticamente o, addirittura, si ferma.

Prof. Montemurro

Linea dedicata

Linee dedicate (o Circuito Diretto Numerico CDN): consente la connessione fissa tra due sedi aziendali, oppure tra la sede centrale e le filiali; tali linee garantiscono sicurezza ed elevate prestazioni.

Sicurezza informatica (o cybersicurezza): insieme delle misure atte a garantire la disponibilità, l'integrità (dati originali, cioè non letti/modificati da persone non autorizzate) (informazione originale, non corrotta), e la riservatezza (informazioni comprensibili solo per chi è autorizzato) delle informazioni gestite dai sistemi informatici.

Svantaggio delle linee dedicate: costi di gestione elevatissimi.

Prof. Montemurro

Linea dedicata



Prof. Montemurro

Reti Informatiche Aziendali

Obiettivo: scambiarsi informazioni in modo efficiente.

Gestione di una rete informatica aziendale prevede:

- i. il monitoraggio della stessa;
- ii. l'aggiornamento della stessa;
- iii. l'eventuale ampliamento della stessa;
- iv. la protezione dei dati conservati in essa.

Configurazioni tipiche di una rete informatica aziendale

- 1. Rete LAN per un ufficio
- 2. Rete client/server aziendale

Prof. Montemurro

Reti LAN per un Ufficio

Creare una rete LAN che consenta agli utenti:

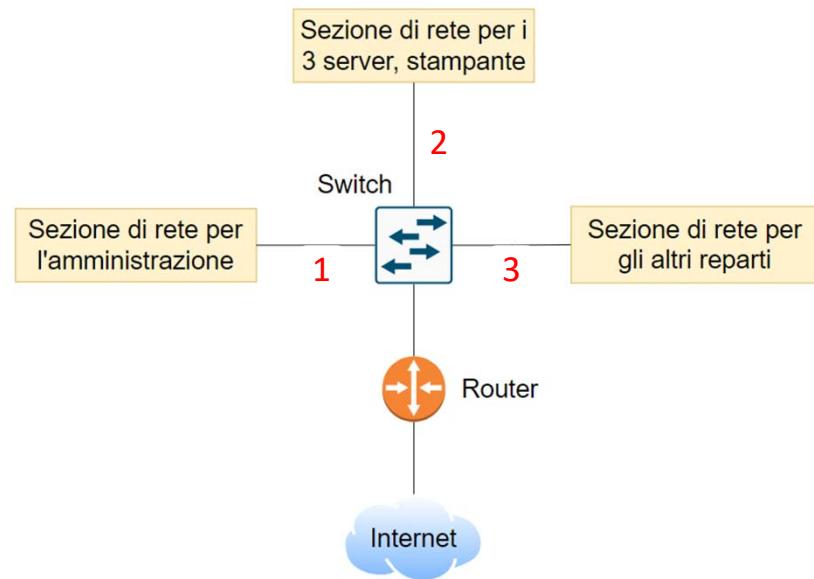
- 1. di comunicare internamente (servirà un server per la gestione della posta elettronica da parte di tutti gli utenti; è il **mail server**);
- 2. di scambiarsi file (servirà un server per la gestione centralizzata degli account degli utenti e dei file; è il **server di dominio**);
- 3. di navigare su internet (servirà un router);
- 4. di stampare documenti cartacei (servirà un **print server** ed una stampante).

Inoltre la rete deve essere suddivisa in due parti:

- i. una parte riservata all'amministrazione e ai tre server;
- ii. una parte per gli altri reparti dell'azienda.

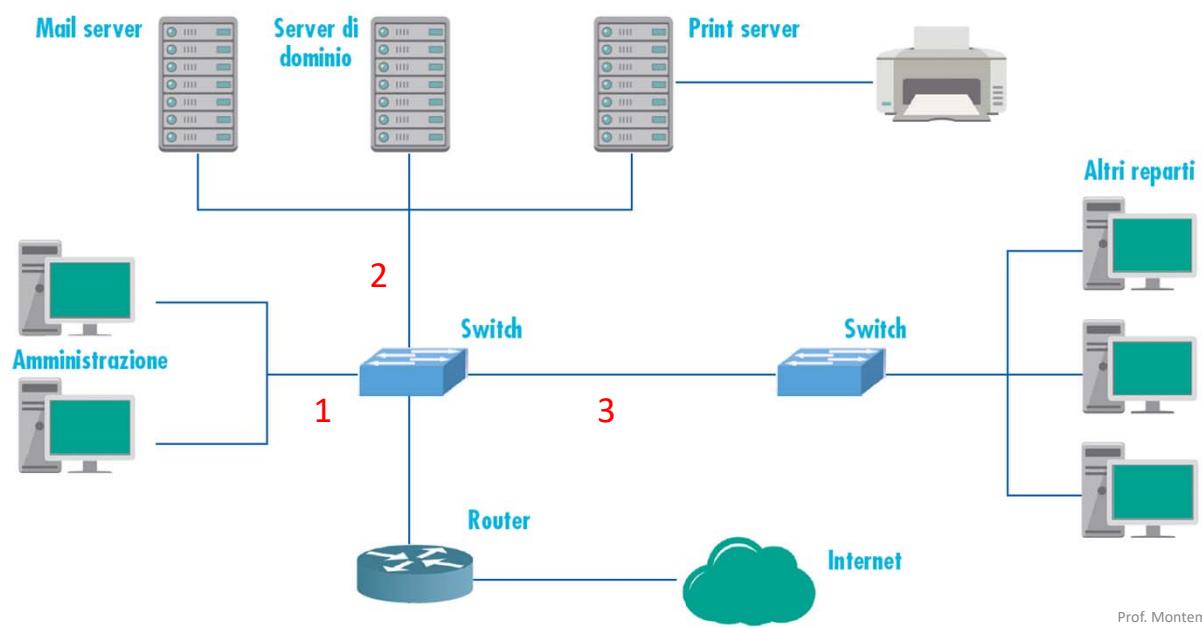
Prof. Montemurro

Reti LAN per un Ufficio: Sezioni di Rete



Prof. Montemurro

Reti LAN per un Ufficio



Prof. Montemurro

Reti Client/Server Aziendale

Rete client/server aziendale: in soldoni è come la rete LAN appena vista, in più c'è una parte della rete LAN accessibile da persone esterne all'azienda (fornitori, clienti ecc.); questa parte della rete LAN è accessibile in extranet.

Extranet: rete informatica interna a un'organizzazione (per es. un'azienda), o porzione di essa, che viene messa a disposizione anche di alcuni soggetti esterni (clienti, fornitori, partner commerciali ecc.) con un accesso regolato da password.

Prof. Montemurro

Reti Client/Server Aziendale

Vulnerabilità: debolezza in un sistema IT che può essere sfruttata da un aggressore per sferrare un attacco di successo.

Firewall di rete: è sempre un software il cui obiettivo è separare internet dalla rete aziendale in quanto quest'ultima è vulnerabile alle intrusioni non autorizzate. Si distinguono due tipi di firewall:

1. **firewall software** i quali sono firewall installati su un computer, cioè su una macchina che non si occupa solo della sicurezza della rete, ma svolge diversi compiti;
2. **firewall hardware** i quali sono software installati su una macchina il cui unico scopo è garantire la sicurezza della rete.



Prof. Montemurro

Reti Client/Server Aziendale

Politiche di default per i servizi web configurabili per il firewall

- Politica default deny** (è quella raccomandata): assume che tutto il traffico sia maligno per cui, di default, tutto è proibito. Quando si identifica del traffico benigno, si aggiunge una **regola di eccezione** per consentirgli l'accesso alla rete.

Vantaggio: questa politica permettere ai responsabili della sicurezza di rete di concentrarsi sul traffico benigno, invece di dover attenzionare tutte le possibili attività maligne.

- Politica default allow:** assume che la maggior parte del traffico sia benigno per cui, di default, tutto è concesso. Quando si identifica del traffico maligno, si aggiunge una **regola di eccezione** per bloccare il suo accesso l'accesso alla rete.

Svantaggio: questa politica obbliga i responsabili della sicurezza di rete ad attenzionare continuamente tutte le possibili attività maligne.

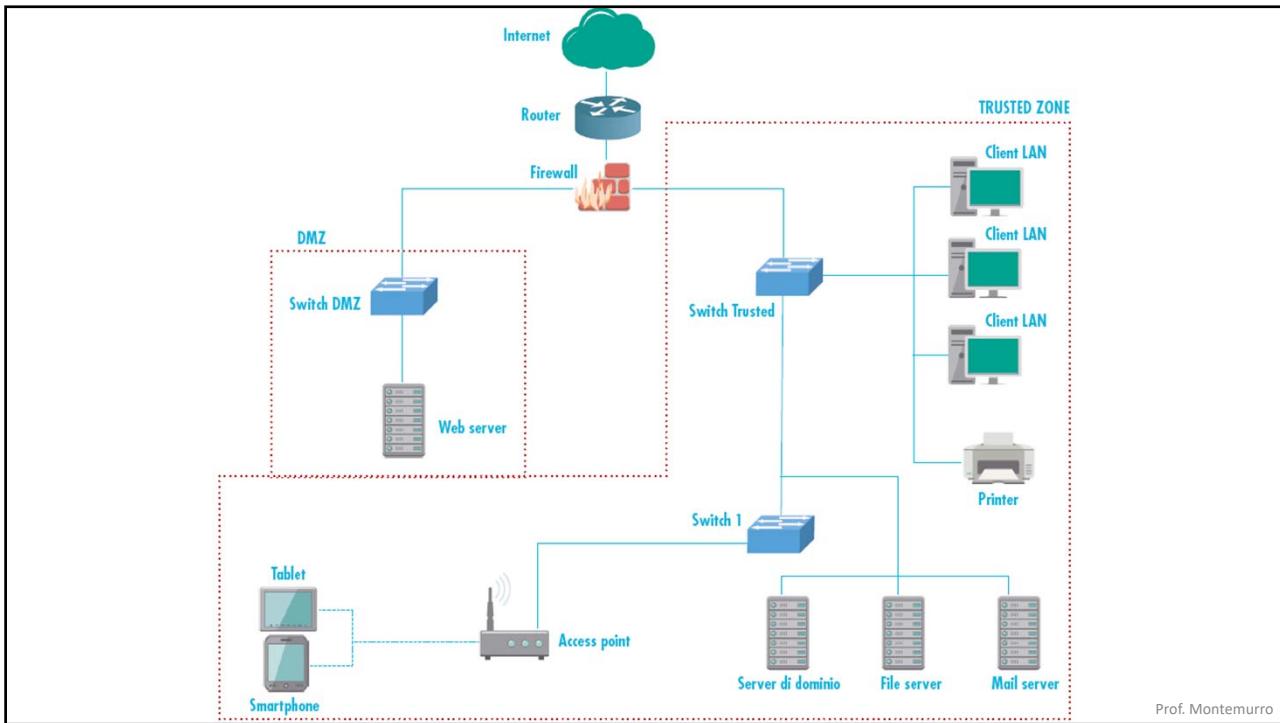
Prof. Montemurro

Reti Client/Server Aziendale

La rete aziendale client/server, per motivi di sicurezza perimetrale, deve essere suddivisa in due grosse sezioni di rete usando il firewall (es. router mette a disposizione la funzionalità di firewall):

- la **trusted zone** la quale non è accessibile direttamente dall'esterno, ma indirettamente attraverso il firewall che bloccherà ogni tentativo di accesso non autorizzato. I dispositivi di rete degli utenti nella trusted zone possono accedere a tale zona autenticandosi con le proprie credenziali le quali sono gestite dal server di dominio;
- la **DeMilitarized Zone** (o **DMZ**) la quale consente solo connessioni verso l'esterno; inoltre, per maggiore sicurezza, si fa in modo che gli host nella DMZ non possano connettersi alla trusted zone.

Prof. Montemurro



Sicurezza nei sistemi informatici

Classe 5ASI
ITCG Fermi

Prof. Montemurro

Definizioni

- **Riservatezza (o confidenzialità, o segretezza) dei dati:** dati leggibili e comprensibili solo dalle persone autorizzate
- **Integrità dei dati:** dati letti e/o modificati solo da persone autorizzate
- **Disponibilità dei dati:** dati devono essere disponibili in qualunque momento per le persone autorizzate per cui occorre garantire la continuità del servizio
- **Paternità (o non ripudiabilità) dei dati:** ogni dato deve essere associato ad un utente che non può ripudiare i dati da lui spediti e/o firmati.
- **Autenticazione ("o" autenticità):** processo di riconoscimento delle credenziali dell'utente per assicurarsi dell'identità di chi invia e/o esegue operazioni.
Metodi di verifica dell'identità di un utente: (1) informazioni riservate (es. password), (2) oggetti elettronici (es. smart card), (3) strumenti di riconoscimento biometrici (es. impronta digitale, fondo retina ecc.)
- **Autorizzazione:** per l'utente autenticato occorre stabilire l'insieme delle autorizzazioni (azioni permesse, risorse accessibili, dati consultabili e/o modificabili)

Prof. Montemurro

Sicurezza Informatica (o Cybersicurezza)

ENISA (Agenzia dell'Unione europea per la cibersicurezza):

Is there a need for a definition? Cybersecurity is an enveloping term and it is not possible to make a definition to cover the extent of the things Cybersecurity covers. Therefore, a contextual definition, based

Nonostante non esista una definizione comunemente accettata, tutte le definizioni di cybersicurezza condividono più o meno gli stessi aspetti.

Cybersicurezza (o sicurezza informatica): insieme delle misure atte a garantire la disponibilità, l'integrità, e la riservatezza, la paternità delle informazioni gestite dai sistemi informatici, e atte a garantire l'autenticazione delle credenziali dell'utente, e le autorizzazioni per l'utente autenticato.

Prof. Montemurro

Definizioni

Minaccia informatica: qualsiasi circostanza, azione, evento intenzionale (attacchi informatici) o accidentale (errori e/o malfunzionamenti) che può causare la perdita di almeno una proprietà di sicurezza (riservatezza, integrità, disponibilità, paternità, autenticazione, autorizzazione).

Esempi di eventi intenzionali ed accidentali

1. **Eventi intenzionali** (o attacchi informatici) (IP spoofing; packet sniffing; connection hijacking; DoS; DDoS)
2. **Eventi accidentali** (o errori e/o malfunzionamenti): (i) inadeguatezza delle strumentazioni, delle politiche, e delle tecnologie di backup; (ii) locale server sensibile alle inondazioni; (iii) armadi contenenti i supporti magnetici/ottici non ignifughi; (iv) errata gestione delle password; (v) mancanza di gruppi di continuità.

Prof. Montemurro

Tipi di Minacce Informatiche

1. **Minacce informatiche naturali:** eventi impossibili da impedire e prevenire quali **calamità naturali** (tempeste, inondazioni, fulmini, incendi, terremoti), **atti vandalici, guerre, sommosse popolari, attacchi terroristici**.
2. **Minacce informatiche umane:** attacchi informatici messi in atto da soggetti (i) che hanno interessi personali ad acquisire le informazioni di un'azienda/soggetto, o (ii) che vogliono limitare l'operatività delle organizzazioni danneggiando i normali processi aziendali (**slide seguente**).
Attacco informatico: tentativo di accesso non autorizzato ad un sistema informativo.

Nel seguito ci concentreremo sulle minacce informatiche umane.

Prof. Montemurro

Definizioni

Processo aziendale: insieme di attività tra loro correlate (nello spazio e nel tempo), svolte secondo una determinata sequenzialità e/o simultaneità. Tale insieme di attività ha un punto di partenza ed un punto di arrivo rappresentato da un risultato misurabile (prodotto o servizio) che contribuisce al raggiungimento della missione dell'organizzazione.

Procedura aziendale: insieme delle regole che consentono di eseguire e portare a termine un processo aziendale.

Prof. Montemurro

Definizioni

Vulnerabilità informatica: debolezza in un sistema informativo. Le vulnerabilità informatiche possono essere sia organizzative e di processo che tecniche, spesso in combinazione tra loro (tali vulnerabilità possono essere sfruttati da attaccanti per effettuare azioni malevoli).

1. **Vulnerabilità organizzative e di processo:** sono riconducibili alla mancata o non corretta definizione o implementazione di misure di sicurezza volte alla tutela della **riservatezza, integrità e disponibilità** delle informazioni.
2. **Vulnerabilità tecniche:** sono dovute a falle di sicurezza del software applicativo, del firmware, dell'hardware ovvero dei protocolli di comunicazione, dovuti principalmente a bug o non corrette configurazioni.

Prof. Montemurro

Definizioni e Prima Classificazione degli Attacchi

2. Minacce informatiche umane

Hacker: persona che studia ed analizza il sistema informatico allo scopo "benefico" di conoscerne e sfruttarne tutte le potenzialità.

Cracker: soggetto che penetra nei sistemi in modo non autorizzato violando i sistemi di protezione.

Prima classificazione degli attacchi informatici

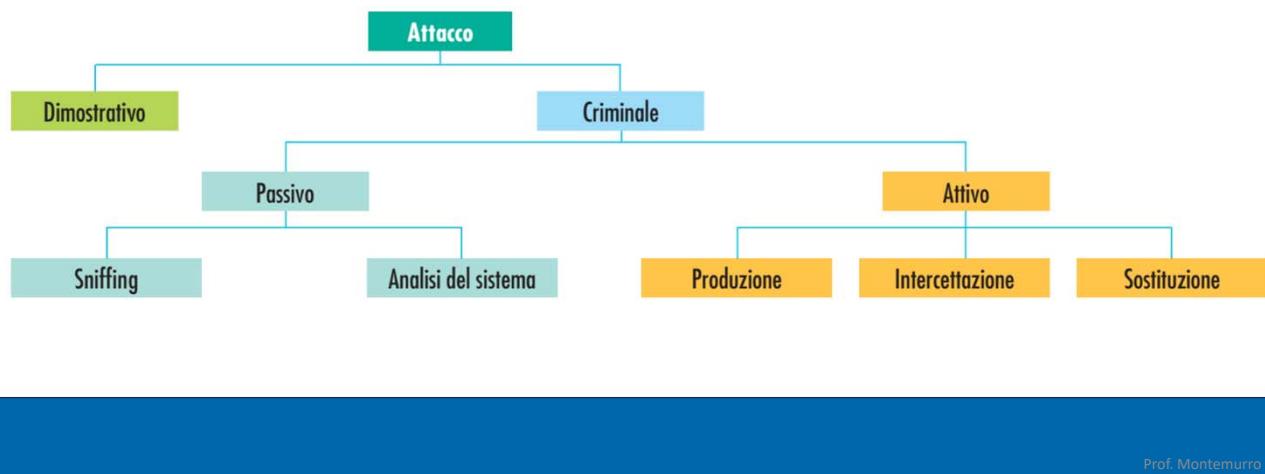
- i. **Attacchi informatici interni:** causati da personale interno (o ex interno) all'organizzazione (es. dipendenti o ex). Questi sono i più insidiosi perché i dipendenti conoscono il sistema informativo, i sistemi di sicurezza, ed hanno le autorizzazioni per accedere al fine di sottrarre direttamente informazioni, e di inserire eventualmente software malevolo in grado di provocare danni, e trasmettere le informazioni all'esterno del sistema informativo.
- ii. **Attacchi informatici esterni:** causati da soggetti esterni all'organizzazione

Prof. Montemurro

Seconda Classificazione degli Attacchi

2. Minacce informatiche umane

Seconda classificazione degli attacchi informatici



Prof. Montemurro

Seconda Classificazione degli Attacchi

2. Minacce informatiche umane

Seconda classificazione degli attacchi informatici

- Attacchi informatici dimostrativi:** non pericolosi, volti a dimostrare l'abilità del cracker
- Attacchi informatici criminali:** volti a intercettare e/o modifica di dati non propri, e/o ad impedire l'utilizzo di determinati servizi agli utenti
 - Attacchi informatici criminali passivi:** non altera sistemi o dati, si limita a spiare i dati (es. **packet sniffing**), o ad eseguire l'analisi del sistema e l'analisi del traffico di rete senza analizzare i contenuti (difficili da rilevare: non producono effetti immediatamente visibili)
 - Attacchi informatici criminali attivi:** attacco a un protocollo di comunicazione sicuro in cui l'aggressore trasmette dati al richiedente, o al fornitore di servizi di credenziali (CSP), o al verificatore, o alla parte affidabile (RP).

Prof. Montemurro

Seconda Classificazione degli Attacchi

2. Minacce informatiche umane

Principali tipi di attacchi informatici criminali attivi

- i. Intercettazione attiva
- ii. Sostituzione di un host
- iii. Produzione
- iv. Email bombing e spamming infetto
- v. Altri (phishing, ransomware ecc.)

Prof. Montemurro

Principali tipi di attacchi informatici criminali attivi

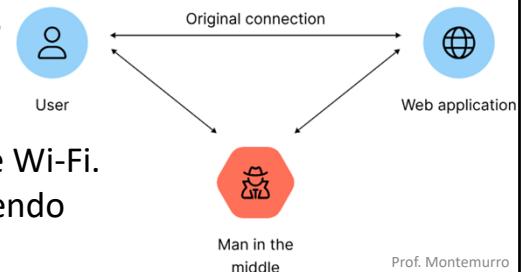
Intercettazione attiva: attacco per intercettare le password al fine di accedere al sistema con l'intento di modificare i dati.

Esempio

Man-in-the-Middle (MitM): sono una famiglia di attacchi tutti accomunati dal fatto che all'inizio c'è una connessione legittima tra due dispositivi, poi un terzo soggetto si mette illegittimamente nel mezzo per cui è in grado di intercettare e modificare i dati in transito.

Attacco MitM basato su falso Access Point

Tale attacco funziona quasi sempre: l'attaccante crea un falso Access Point, spesso dal nome simile ma non uguale a quello legittimo, per creare un ponte tra l'utente e il router della rete Wi-Fi. La gente si connette all'Access Point falso e apprendo così le porte del suo dispositivo all'hacker.



Prof. Montemurro

Principali tipi di attacchi informatici criminali attivi

Sostituzione di un host: sostituzione di un dispositivo legittimo con uno malevolo per impersonarlo nella rete.

Esempio

Premessa: a ogni scheda di rete viene assegnato un **indirizzo IP** (o **indirizzo logico**) il quale identifica univocamente la scheda di rete.

IP spoofing: tecnica per eseguire vari tipi di attacchi tra cui anche la sostituzione di un host (anche per il MitM). Tale tecnica consiste nel falsificare l'indirizzo IP del mittente per far sembrare che il traffico provenga da un host legittimo. Fatto ciò, l'attaccante può intercettare e modificare i dati a cui ha accesso l'host del mittente (es. il DB dell'azienda).

Prof. Montemurro

Principali tipi di attacchi informatici criminali attivi

Produzione: malintenzionati producono nuovi componenti software e li inseriscono nel sistema informatico con lo scopo di causare un danno, non per prelevare le informazioni (atti di sabotaggio). L'obiettivo è ridurre l'integrità e la disponibilità dei dati.

Esempi

1. Attacchi con **malware** (**malicious software**)

- i. Attacchi con virus **virus**: programmi che infettano programmi esistenti (**programma ospite**); i virus provocano danni e si replicano infettando altri host durante il trasferimento del file infetto da un host ad un altro
- ii. Attacchi con **worm**: a differenza dei virus, sono programmi autonomi che non si attaccano ad altri programmi, ma si propagano mediante diffusione dentro reti di computer o tramite email; i worm, come i virus, provocano danni e si replicano infettando altri host

Prof. Montemurro

Principali tipi di attacchi informatici criminali attivi

Esempi

1. Attacchi con **malware** (**malicious software**)
 - iii. Attacchi **Denial of Service DoS**: far eseguire all'host operazioni inutili per tenerlo occupato così da impedirgli di offrire i propri servizi
 - iv. **Trojan horse** ([slide successiva](#))
 - v. **Ransomware** (**ransom software**): malware che blocca in modo permanente l'accesso ai dati o ai dispositivi finché il proprietario dei dati non paga un riscatto. La maggior parte delle volte i ransomware sono trojan horse.

Prof. Montemurro

Principali tipi di attacchi informatici criminali attivi

Email bombing: invio di enormi quantità di messaggi email per mandare in crash il server di posta (è un esempio di DoS).

Spamming infetto: invio di messaggi email contenenti link dannosi per infettare il dispositivo della vittima o rubare dati sensibili.

Esempio

Trojan horse (o **cavallo di troia**): è un malware nascosto all'interno di un altro software apparentemente utile (es. videogiochi). L'obiettivo è assumere il controllo del computer; non si installa automaticamente come i virus. I trojan possono installare backdoor, o keylogger, oppure inviare messaggi di spam.

Keylogger: hardware o software che intercetta tutto quello che viene digitato sulla tastiera del computer.

Prof. Montemurro

Principali tipi di attacchi informatici criminali attivi

Phishing: attività illegale che ha come scopo il furto di identità e di dati sensibili tramite le comunicazioni elettroniche (es. email; SMS ecc.), sfruttando tecniche di **ingegneria sociale** per ingannare le vittime.

Schema standard di attacco

1. L'attaccante invia alla vittima un messaggio e-mail che simula quello di un'istituzione nota al destinatario (in genere ci sono errori ortografici).
2. Il messaggio ha toni allarmistici e richiede un intervento urgente.
3. La vittima è invitata a seguire un link presente nella mail.
4. Tale link non punta al vero sito ma ad una copia fittizia identica, in cui viene chiesto di compilare un form con dati riservati.

Prof. Montemurro

Principali tipi di attacchi informatici criminali attivi

Ingegneria sociale: nell'ambito della sicurezza informatica, complesso di strategie e metodi di manipolazione psicologica e di persuasione volti a indurre un utente a rivelare informazioni riservate (dati personali, credenziali di accesso, numeri di carte di credito, di conti bancari, di previdenza sociale, ecc.).

Prof. Montemurro

Sicurezza nei Sistemi Informativi

Problema: la rete aziendale, essendo connessa ad internet, è soggetta ad attacchi informatici.

Obiettivo: proteggere le informazioni importanti e riservate in quanto esse rappresentano la risorsa più importante di ogni organizzazione. Per farlo si applica il **principio minimo di sicurezza** in base al quale bisogna:

1. proteggersi dagli attacchi informatici passivi;
2. riconoscere gli attacchi informatici attivi.

In pratica, bisogna analizzare tutte le componenti del sistema, e individuare per ciascuna di esse tutte le tecniche di attacco passive e attive.

Prof. Montemurro

Sicurezza nei Sistemi Informativi

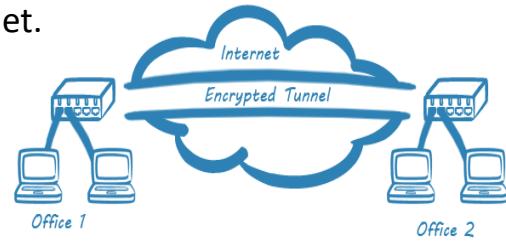
Pilastri della sicurezza

1. **Prevenzione:** mediante protezione dei sistemi e delle comunicazioni con crittografia, firewall, VPN ecc.
2. **Rilevazione:** mediante il monitoraggio ed il controllo degli accessi tramite autenticazione con password, e certificati
3. **Investigazione:** mediante l'analisi dei dati, il controllo interno grazie al confronto ed alla collaborazione degli utenti ecc. (es. si pongono domande agli utenti per portare avanti un'indagine)

Prof. Montemurro

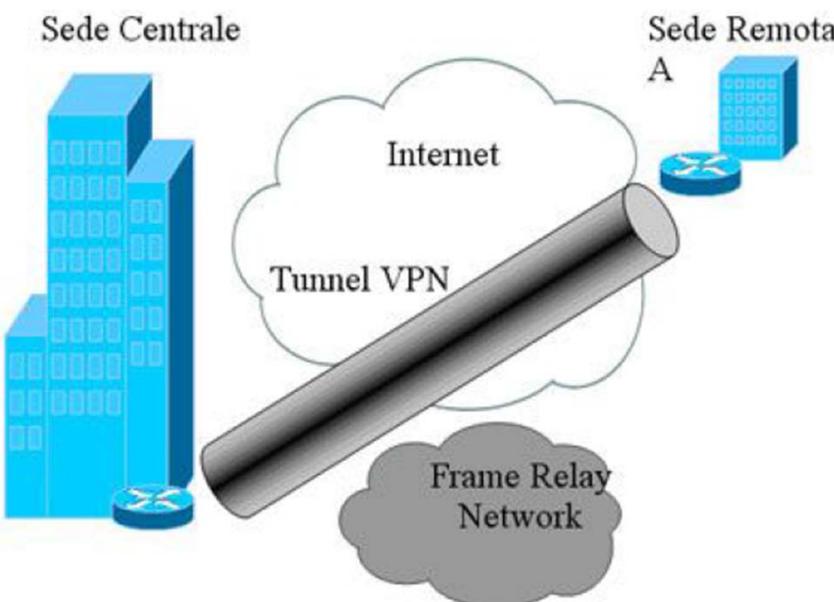
Sicurezza nei Sistemi Informativi: Misure di Prevenzione

1. **Crittografia:** garantisce riservatezza e integrità dei dati, previene attacchi informatici passivi di tipo packet sniffing
2. **Autenticazione degli utenti:** processo di riconoscimento delle credenziali dell'utente per assicurarsi dell'identità di chi invia e/o esegue operazioni
3. **Firewall:** vedi slide su fondamenti di networking
4. **Linee dedicate (o reti private):** vedi slide su fondamenti di networking
5. **Reti private virtuali (o Virtual Private Network) VPN:** è una rete informatica cifrata, privata che si appoggia su una rete di telecomunicazioni **pubblica** (ecco perché virtuale) come internet.
E' come se all'interno della rete **pubblica** ci fosse un **tunnel** sicuro tra le due reti informatiche che vogliamo collegare.



Prof. Montemurro

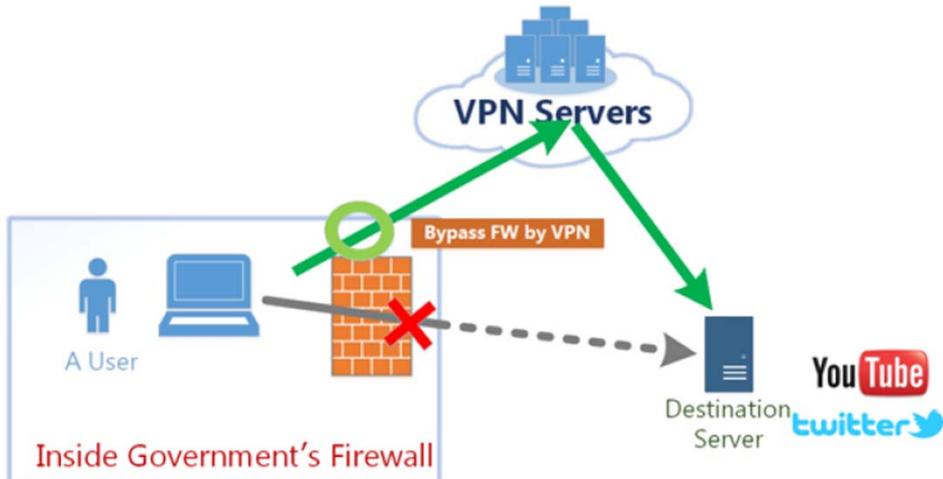
Sicurezza nei Sistemi Informativi: Misure di Prevenzione



Prof. Montemurro

Sicurezza nei Sistemi Informativi: Misure di Prevenzione

Curiosità: VPN per bypassare le georestrizioni.



Prof. Montemurro

Sicurezza nei Sistemi Informativi: Misure di Prevenzione

6. **Posizionamento dei server in locali protetti** (armadi, locali accessibili solo dai tecnici autorizzati, luoghi lontani dalla sede aziendale come i data center il cui accesso è spesso protetto sia in modo fisico che virtuale).
Data center: insieme di server posti in un unico luogo al fine di favorire una centralizzazione (i) della gestione, (ii) della sicurezza, (iii) della manutenzione dei server stessi.
7. **Sistemi di alimentazione autonoma tramite gruppi di continuità**
8. **Backup periodico**, ossia salvataggio periodico dei dati

Prof. Montemurro



Prof. Montemurro

Normativa sulla Sicurezza e sulla Privacy

Classe 5ASI
ITCG Fermi

Prof. Montemurro

Definizioni

- **Riservatezza (o confidenzialità, o segretezza) dei dati:** dati leggibili e comprensibili solo dalle persone autorizzate
- **Integrità dei dati:** dati letti e/o modificati solo da persone autorizzate
- **Disponibilità dei dati:** dati devono essere disponibili in qualunque momento per le persone autorizzate per cui occorre garantire la continuità del servizio
- **Paternità (o non ripudiabilità) dei dati:** ogni dato deve essere associato ad un utente che non può ripudiare i dati da lui spediti e/o firmati.
- **Autenticazione ("o" autenticità):** processo di riconoscimento delle credenziali dell'utente per assicurarsi dell'identità di chi invia e/o esegue operazioni.
Metodi di verifica dell'identità di un utente: (1) informazioni riservate (es. password), (2) oggetti elettronici (es. smart card), (3) strumenti di riconoscimento biometrici (es. impronta digitale, fondo retina ecc.)
- **Autorizzazione:** per l'utente autenticato occorre stabilire l'insieme delle autorizzazioni (azioni permesse, risorse accessibili, dati consultabili e/o modificabili)

Prof. Montemurro

Azioni per la Sicurezza Informatica di un'Azienda

1. Definizione e attuazione di politiche che garantiscano le sei proprietà della sicurezza elencate nella slide precedente
2. Identificazione delle informazioni critiche e strategiche
3. Individuazione delle possibili minacce e delle strategie operative da attuare per prevenirle
4. Analisi della normativa vigente e conseguente adeguamento dei SIA
5. Analisi economica dei costi per la messa in opera degli interventi di adeguamento individuati e scelta delle alternative più convenienti

Prof. Montemurro

Giurisprudenza Informatica

1. Leggi sui crimini informatici e relative sanzioni per i malintenzionati e per coloro che eseguono azioni illecite
2. Leggi su **misure minime di sicurezza** che coloro che trattano i dati devono adottare (i) per ridurre al minimo i rischi di distruzione o perdita anche parziale dei dati, e (ii) per tutelare la privacy di tutti coloro che hanno fornito dati personali. Ciò viene fatto tramite l'utilizzo:
 - i. di sistemi di autenticazione;
 - ii. di sistemi di autorizzazione;
 - iii. di soluzioni antivirus;
 - iv. di sistemi di protezione da intrusioni maligne;
 - v. di soluzioni di sicurezza per prevenire/evitare la commissione di reati da parte dei dipendenti (azienda può controllare il PC di un dipendente per prevenire gli abusi, ma non per monitorare la prestazione lavorativa).

Prof. Montemurro

Giurisprudenza Informatica

3. Legge n. 675/1996:

- i. istituisce il **garante della privacy** il quale è un organo composto da 4 membri eletti dal Parlamento, che rimangono in carica per un mandato di 4 anni rinnovabile. Il garante è un'autorità indipendente il cui obiettivo è assicurare (a) la tutela dei diritti e delle libertà fondamentali, (b) il rispetto della dignità nel trattamento dei dati personali, e (c) diffondere la cultura della tutela dei dati personali;
- ii. definisce il **titolare del trattamento dei dati** (cioè colui che usa i dati) come la persona fisica/giuridica, la Pubblica Amministrazione, un qualsiasi ente, organismo cui competono le scelte di fondo sulle finalità e sulle modalità di trattamento dei dati.

Prof. Montemurro

Giurisprudenza Informatica

4. Legge sulla privacy (o D.lgs. n. 196/2003):

- i. introduce il diritto alla protezione dei dati personali; il proprietario dei dati non è chi li usa, ma le persone a cui i dati si riferiscono;
- ii. sancisce, per chi tratta dati personali, (a) il dovere di proteggerli e conservarli, (b) l'obbligo della compilazione annuale del Documento Programmatico sulla Sicurezza (DPS) per attestare la corretta esecuzione e adempimento delle misure atte a proteggere e conservare i dati personali.

Prof. Montemurro

Giurisprudenza Informatica

5. **Regolamento europeo sulla privacy (General Data Protection Regulation GDPR)**: regolamento UE, riguardante la protezione dei dati personali, rivolto a enti e organizzazioni che raccolgono dati personali online, di cittadini europei, e li archiviano su server residenti sia all'interno che all'esterno dell'Unione Europea. Le novità introdotte da tale regolamento riguardano:
- i. le condizioni che regolano la diffusione dei dati personali; il trattamento dei dati dell'utente deve essere richiesto in modo esplicito, tracciabile, e non deve condizionare l'accesso ad un servizio;
 - ii. il diritto alla cancellazione dei dati personali (diritto all'oblio), cioè il proprietario dei dati personali può chiedere al titolare del trattamento degli stessi di cancellarli.

Prof. Montemurro

Giurisprudenza Informatica

5. **Regolamento europeo sulla privacy (General Data Protection Regulation GDPR)**: istituisce il **registro del trattamento dei dati personali** il quale prevede che il titolare del trattamento dei dati personali:
- i. conservi la documentazione di tutti i trattamenti effettuati sotto la propria responsabilità;
 - ii. indichi obbligatoriamente, per ogni trattamento eseguito, tutte le informazioni che comprovano la conformità di ciascuna operazione alle disposizioni del regolamento GDPR.

Prof. Montemurro



Prof. Montemurro

Principi di Crittografia

Classe 5ASI
ITCG Fermi

Prof. Montemurro

Definizioni

- **Riservatezza (o confidenzialità, o segretezza) dei dati:** dati leggibili e comprensibili solo dalle persone autorizzate
- **Integrità dei dati:** dati letti e/o modificati solo da persone autorizzate
- **Disponibilità dei dati:** dati devono essere disponibili in qualunque momento per le persone autorizzate per cui occorre garantire la continuità del servizio
- **Paternità (o non ripudiabilità) dei dati:** ogni dato deve essere associato ad un utente che non può ripudiare i dati da lui spediti e/o firmati.
- **Autenticazione ("o" autenticità):** processo di riconoscimento delle credenziali dell'utente per assicurarsi dell'identità di chi invia e/o esegue operazioni.
Metodi di verifica dell'identità di un utente: (1) informazioni riservate (es. password), (2) oggetti elettronici (es. smart card), (3) strumenti di riconoscimento biometrici (es. impronta digitale, fondo retina ecc.)
- **Autorizzazione:** per l'utente autenticato occorre stabilire l'insieme delle autorizzazioni (azioni permesse, risorse accessibili, dati consultabili e/o modificabili)

Prof. Montemurro

Crittografia

Problema: le reti, per loro natura, non sono sicure, infatti basta un **analizzatore di rete** (o **packet sniffer**) come **Wireshark** per intercettare le informazioni (per fortuna cifrate) che viaggiano su una rete.

Crittografia (o **criptografia**, o **crittaggio**, o **criptaggio**, o **cifratura**, o **codifica**, o **codificazione**): significa *scrittura nascosta*, e riguarda i metodi per rendere un messaggio non comprensibile a persone che non sono autorizzate a leggerlo.

Algoritmo di codifica (o **a. di cifratura**, o **a. di criptazione**): metodo per trasformare i simboli in chiaro, i quali compongono il messaggio in chiaro, in simboli cifrati i quali compongono il messaggio cifrato.

Algoritmo di decodifica (o **algoritmo di decifratura**): metodo per trasformare i simboli cifrati in simboli in chiaro.

Prof. Montemurro

Crittografia



Prof. Montemurro

Crittografia

Approfondimento

Differenza tra codifica e cifratura in crittografia

- **Codifica:** metodo che consiste nel sostituire alcune parole con altre
- **Cifratura:** metodo che consiste nel sostituire lettere o caratteri

Cifrario: sistema convenzionalmente stabilito, per tradurre il linguaggio chiaro in linguaggio segreto, comprensibile soltanto a chi sia a conoscenza della convenzione

Esempio

Vogliamo inviare la parola AIUTO.

- Metodo di codifica: invece di trasmettere AIUTO, trasmetto HELP, cioè sostituisco completamente la parola AIUTO con un'altra
- Metodo di cifratura: invece di trasmettere AIUTO, trasmetto BLVUP; in questo caso ciascuna lettera è stata sostituita con quella che la segue nell'alfabeto.

Prof. Montemurro

Cifratura Simmetrica e Asimmetrica

Crittografia si basa su due elementi fondamentali:

1. l'algoritmo di codifica;
2. le **chiavi (o parametri)**; nell'esempio precedente, la chiave è la posizione del carattere sostitutivo (un carattere in avanti nell'alfabeto se inviamo BLVUP invece di AIUTO).

Chiave di cifratura + algoritmo di cifratura: servono per trasformare un messaggio in chiaro in un messaggio cifrato.

Chiave di decifratura + algoritmo di decifratura: servono per trasformare un messaggio cifrato in un messaggio in chiaro.

Prof. Montemurro

Cifratura Simmetrica e Asimmetrica

Cifratura simmetrica: chiave di cifratura e chiave di decifratura coincidono; in questo caso si parla di **chiave comune**.



Prof. Montemurro

Cifratura Simmetrica e Asimmetrica

Cifratura asimmetrica: chiave di cifratura e chiave di decifratura non coincidono; in questo caso la chiave di cifratura è chiamata **chiave pubblica** (o **Public Key**) **PK** in quanto è disponibile per chiunque, e la chiave di decifratura è chiamata **chiave privata** (o **private key**, o **Secret Key**) **SK** la quale è posseduta solo dal destinatario del messaggio cifrato.

Il destinatario (non il mittente) genera la coppia di chiavi (pubblica e privata); la chiave privata la tiene per sé, mentre la chiave pubblica la rende nota a tutti, compreso il potenziale mittente del messaggio. Il mittente userà la chiave pubblica inviatagli dal destinatario per criptare il messaggio; tale messaggio potrà essere decriptato solo dal destinatario in quanto solo lui possiede la chiave privata.

Prof. Montemurro

Cifratura Simmetrica e Asimmetrica

Esempio

Il destinatario compra un lucchetto, la chiave se la tiene, e manda il lucchetto aperto al mittente. Il mittente scrive il messaggio, lo chiude in una scatola di ferro che poi sigilla col lucchetto inviatogli dal destinatario. Dunque tale scatola può essere aperta solo dal destinatario in quanto solo lui ha la chiave del lucchetto.



Prof. Montemurro

Cifratura Simmetrica e Asimmetrica

Problema della cripitura asimmetrica: anche se il destinatario può verificare che nessuno abbia letto o modificato il messaggio durante il suo viaggio lungo la rete, purtroppo il destinatario non può verificare la paternità del messaggio, cioè non può avere la certezza del mittente. Ciò perché la chiave pubblica è per l'appunto pubblica per cui chiunque può usarla per cifrare un messaggio ed inviarlo al destinatario spacciandosi per un mittente noto al destinatario (es. Bob è amico di Alice; Alice condivide la chiave pubblica e si tiene la chiave privata; può capitare che un'altra persona usi la chiave pubblica per cifrare un messaggio e inviarlo ad Alice spacciandosi per Bob).

Soluzione per avere non ripudiabilità del mittente e del destinatario: cripitura a doppia codifica.

Prof. Montemurro

Cifratura a Doppia Codifica

Premessa: il mittente A genera una coppia di chiavi, la chiave privata A e la chiave pubblica A. Il destinatario B genera un'altra coppia di chiavi, la chiave privata B e la chiave pubblica B.

1. Il mittente A codifica il messaggio che vuole inviare a B con la propria chiave privata A (il contrario della cifratura asimmetrica); questa codifica garantisce la non ripudiabilità del mittente A, in quanto il destinatario B può leggere il messaggio di A solo utilizzando la chiave pubblica che il mittente A gli ha inviato (ricorda che solo la chiave pubblica che è stata generata insieme alla chiave privata può decodificare i messaggi cifrati con la chiave privata stessa).
2. Successivamente il mittente A sottopone il messaggio ad una seconda codifica usando la chiave pubblica B generata dal destinatario B. Ciò garantisce la riservatezza del messaggio in quanto solo il destinatario B è in possesso della chiave privata B in grado di decifrare il messaggio.

Prof. Montemurro

Cifratura a Doppia Codifica

3. Il destinatario B decodifica il messaggio usando la propria chiave privata B. Ciò garantisce:
 - i. la riservatezza del messaggio (già detto prima);
 - ii. la non ripudiabilità del destinatario B, cioè B non può negare di aver ricevuto il messaggio (prima abbiamo parlato della non ripudiabilità del mittente A) "in quanto" solo il destinatario B è in possesso della chiave privata B in grado di decifrare il messaggio inviatogli da A.
4. Successivamente il destinatario B sottopone il messaggio ad una seconda decodifica usando la chiave pubblica che il mittente A gli ha inviato. Ciò garantisce la non ripudiabilità del mittente A (già detto).

Prof. Montemurro

Cifratura a Doppia Codifica



Prof. Montemurro

Crittoanalisi

Crittoanalisi (o criptoanalisi): metodi di ricostruzione del testo in chiaro a partire da uno o più testi cifrati di cui non si possiede la chiave.

Criptosistema (o crittosistema, o sistema criptografico, o sistema crittografico, o sistema di cifratura): è una quintupla costituita da (1) algoritmo di codifica, (2) algoritmo di decodifica, (3) testo in chiaro, (4) testo cifrato, e (5) la chiave.

Principio di Kerckhoffs: la sicurezza di un crittosistema deve dipendere solo dalla segretezza della chiave, e non dalla segretezza dell'algoritmo di codifica e decodifica usato (in genere l'algoritmo è noto in quanto è uno standard).

Obiettivo dei cracker: individuare la chiave (visto che l'algoritmo è in genere noto).

Prof. Montemurro

One-Time Pad

One-Time Pad: è chiamato cifrario assolutamente sicuro; mittente e destinatario, per comunicare in modo segreto, condividono un blocco di testo (**pad**) costituito da una sequenza di lettere casuali, lunga quanto il messaggio da inviare. Questa sequenza viene condivisa segretamente in anticipo tra le due parti, cioè prima della comunicazione vera e propria.

La chiave cambia per ogni lettera, nel senso che non si usa la stessa regola di sostituzione per l'intero messaggio, ma una chiave diversa per ogni carattere del messaggio.

Esempio

Il messaggio CIAO viene codificato in ZUCY usando la chiave XMCK. Si considera A = 0, B = 1, C = 2 e così via. La lettera C viene codificata usando la chiave X (X = 23) per cui $2 + 23 = 25 = Z$. Per decifrare:

$$\text{numeroLetteraCodificata} - \text{numeroChiave} = 25 - 23 = 2 = C$$

Prof. Montemurro



Prof. Montemurro

Autenticazione dell'Utente

Classe 5ASI
ITCG Fermi

Prof. Montemurro

Definizioni

- **Riservatezza (o confidenzialità, o segretezza) dei dati:** dati leggibili e comprensibili solo dalle persone autorizzate
- **Integrità dei dati:** dati letti e/o modificati solo da persone autorizzate
- **Disponibilità dei dati:** dati devono essere disponibili in qualunque momento per le persone autorizzate per cui occorre garantire la continuità del servizio
- **Paternità (o non ripudiabilità) dei dati:** ogni dato deve essere associato ad un utente che non può ripudiare i dati da lui spediti e/o firmati.
- **Autenticazione ("o" autenticità):** processo di riconoscimento delle credenziali dell'utente per assicurarsi dell'identità di chi invia e/o esegue operazioni.
Metodi di verifica dell'identità di un utente: (1) informazioni riservate (es. password), (2) oggetti elettronici (es. smart card), (3) strumenti di riconoscimento biometrici (es. impronta digitale, fondo retina ecc.)
- **Autorizzazione:** per l'utente autenticato occorre stabilire l'insieme delle autorizzazioni (azioni permesse, risorse accessibili, dati consultabili e/o modificabili)

Prof. Montemurro

Metodi di Autenticazione Elettronica

1. **Metodo di autenticazione basato su qualcosa (di immateriale) che l'utente ha** (password, PIN, ecc.)
2. **Metodo di autenticazione basato su qualcosa (di materiale) che l'utente possiede** (carte fisiche o elettroniche, dispositivi hardware, ecc.)
3. **Metodo di autenticazione basato su qualcosa di personale dell'utente** (dati biometrici come le impronte digitali, la conformazione della retina, ecc.)

Prof. Montemurro

Strumenti di Autenticazione Forte

1. **Accesso logico** ([password](#) e [passphrase](#))
2. **OTP** (One-Time Password);
3. **Firma digitale**, [Posta Elettronica Certificata \(PEC\)](#)

Prof. Montemurro

1a Autenticazione con Password

Problemi

1. Memorizzazione delle password da parte dell'amministratore di sistema (o sistemista)
2. Scegliere e ricordarsi la password da parte dell'utente

Soluzioni al problema 1

- i. Memorizzare le password in chiaro in un file protetto (es. file protetto da una password, o file protetto da cifratura); questa soluzione è sconsigliata
- ii. Memorizzare le password in forma cifrata

Prof. Montemurro

1a Autenticazione con Password

Per accedere ad un servizio, per crittografare un documento, serve una **password** (o **parola chiave**, o **parola d'ordine**) oppure una **passphrase** (o **frase d'accesso**, o **frase segreta**) la quale è un insieme di parole o stringhe alfanumeriche.

Problema delle password: sono soggette ad **attacco a dizionario** (o **attacco forza bruta**) il quale si basa su un algoritmo che prova ad usare come password

- tutte le parole del dizionario (in diverse lingue)
- i nomi propri
- le permutazioni di ogni stringa (es. alcune permutazioni di *ciao* sono: *ciao*, *caio*, *cioa*, *coia*, *ocia*, *ocai*, *iaco*, *ioca* ecc.)

Prof. Montemurro

1a Autenticazione con Password

Difese dagli attacchi a dizionario (e soluzioni al problema 2 nella slide 5)

i. **Alterare caratteri maiuscoli e minuscoli**

Esempio: aBRACaDAbRa invece di abracadabra

ii. **Sostituire O con 0, L con 1, S con 5, A con @ in una parola conosciuta**

Esempio: Segret0 invece di segreto

iii. **Deformazione di parole composte**

Esempio: mezzaBotte, mezzaCotte invece di mezzanotte

iv. **Deformazione di parole composte + numeri** (sostituire O con 0, L con 1, S con 5, A con @) e **simboli** (sostituire doppie DD con 2D dove D è un generico carattere ripetuto due volte)

Esempio: me2z@B02te invece di mezzaBotte

v. **Parole e numeri senza senso**

Esempio: innestare data di nascita all'interno del nome ma995rio invece di mario995, sa19ra94 invece di sara1994

Prof. Montemurro

1b Autenticazione con Passphrase

Passphrase: insieme di parole o stringhe alfanumeriche.

Problema: scegliere e ricordarsi la passphrase da parte dell'utente.

Soluzioni

i. **Prendere una frase** (residente nella nostra memoria a lungo termine) e **ridurla a 12 caratteri** (o comunque ad un numero modesto di caratteri).

Esempio: frase "Nel mezzo del cammin di nostra vita" diventa "nLmZdLcM" dopo aver applicato le seguenti 4 trasformazioni:

- eliminare vocali: "Nl mzz dl cmmn d nstr vt"
- eliminare doppie: "Nl mz dl cmn d nstr vt"
- eliminare gli spazi e mantenere solo i primi 12 caratteri: "NlmzdlcM"
- alternare una lettera maiuscola ad una minuscola: " nLmZdLcM"

Nota bene: occorre ricordarsi le trasformazioni per generare la passphrase.

Prof. Montemurro

1b Autenticazione con Passphrase

Soluzioni

- ii. Prendere frasi semplici che includono segni di punteggiatura e maiuscole
Esempio: "A cura di LoRusso (non Lo Russo, Editore)"

- iii. **Metodo Diceware:** utilizzo di 5 dadi per selezionare le parole a caso da un elenco speciale, chiamato **elenco di parole Diceware**, che contiene 7776 brevi parole, abbreviazioni, stringhe di caratteri nelle diverse lingue, facili da ricordare. Da ogni lancio dei 5 dadi si ottiene un numero di 5 cifre che corrisponde ad una parola nell'elenco di parole di Diceware.

Esempio: generazione di una passphrase composta da 3 parole

- Numero "21434" → parola "gatto"
- Numero "53662" → parola "marrone"
- Numero "62352" → parola "penna"

Passphrase: gattomarronepenna.

Prof. Montemurro

1b Autenticazione con Passphrase

Soluzioni

- iv. Concatenare parole brevi che piacciono all'utente numerandole e alternandole in forma maiuscola e minuscola

Esempio: SOLE1pappa2MARE3ciccia4

Prof. Montemurro

2 Autenticazione con OTP

One-Time Password: password generata da dispositivi hardware, temporanea, cioè "usa e getta". Questa è più sicura delle password ideate dall'utente.

Metodi di generazione di OTP

- i. **Lista condivisa di password OTP tra utenti e sistema di autenticazione**
- ii. **Schema di Lamport**
- iii. ...

Prof. Montemurro

2 Autenticazione con OTP

Metodi di generazione di OTP

- i. **Lista condivisa di password OTP tra utenti e sistema di autenticazione** (non viene usata): a ogni utente del sistema viene associata una lista contenente un elenco di password, ciascuna delle quali deve essere utilizzata una sola volta.

Svantaggi:

- a. Manutenzione della lista condivisa; mantenere l'elenco aggiornato su più utenti è impegnativo e incline agli errori
- b. Appropriazioni indebite del file contenente la lista di password OTP
- c. Utente può smarrire l'elenco o non aggiornarlo



Prof. Montemurro

2 Autenticazione con OTP

Metodi di generazione di OTP

ii. **Schema di Lamport** (si usa uno simile a questo) basato su tre ingredienti

- w , chiamato **seme**, è un valore arbitrario, scelto dall'utente, che deve essere mantenuto segreto
- H , chiamata **funzione di hash**, è una funzione unidirezionale, cioè dato w , è facile calcolare $H(w)$, ma, dato $H(w)$, è impossibile calcolare w
- n è un numero naturale che definisce il numero di autenticazioni che devono essere eseguite basandosi sul valore segreto w (es. $n = 100$ significa che con la funzione H si potranno generare 100 password e quindi si potranno eseguire 100 autenticazioni)

Schema di Lamport: algoritmo nel quale, a partire dal valore del seme w associato all'utente, viene generato, mediante funzioni di hash, un insieme di password in sequenza, aventi come seme proprio w .

Prof. Montemurro

2 Autenticazione con OTP

La funzione H è usata per definire la sequenza di password OTP (H^i significa che la funzione H è applicata i volte; es. $H^3(w) = H(H(H(w)))$):

OTP usate dall'utente	Valori di verifica y usati dal sistema di autenticazione
Prima di usare OTP_1 il sistema sa $H^n(w)$	$y_1 = H(OTP_1) = H^n(w)$
$OTP_1 = H^{n-1}(w)$	$y_2 = H(OTP_2) = H^{n-1}(w)$
$OTP_2 = H^{n-2}(w)$:
\vdots	$y_{n-2} = H(OTP_{n-2}) = H^3(w)$
$OTP_{n-2} = H^2(w)$	$y_{n-1} = H(OTP_{n-1}) = H^2(w)$
$OTP_{n-1} = H^1(w) = H(w)$	$y_n = H(OTP_n) = H(w)$
$OTP_n = H^0(w) = w$	

Prof. Montemurro

2 Autenticazione con OTP: Esempio

- All'inizio, cioè prima che l'utente usi la prima password $OTP_1 = H^{n-1}(w)$, il sistema conosce $y_1 = H^n(w)$.
- Quando l'utente fornisce al sistema $OTP_1 = H^{n-1}(w)$, il sistema applica H e ottiene $H(H^{n-1}(w)) = H^n(w)$.
- A questo punto il sistema verifica se l'OTP usato dall'utente è valido eseguendo il confronto di seguito indicato (val. calcolato = val. in memoria):

$$H^n(w) = y_1$$
 e conserva il valore $y_2 = H^{n-1}(w)$.
- Quando l'utente usa la $OTP_2 = H^{n-2}(w)$, il sistema applica H e ottiene $H(H^{n-2}(w)) = H^{n-1}$.
- A questo punto il sistema verifica se l'OTP usato dall'utente è valido eseguendo il confronto di seguito indicato:

$$H^{n-1}(w) = y_2$$
 e conserva il valore $y_3 = H^{n-2}(w)$. E così via...

Prof. Montemurro

2 Autenticazione con OTP

Generalizzando, il confronto che il sistema di autenticazione esegue ogni volta che l'utente fornisce una password OTP è il seguente:

$$H(\text{password OTP corrente}) = \text{password precedente}$$

o equivalentemente:

$$H(OTP_i) = OTP_{i-1}$$

Inoltre il sistema memorizza la password OTP corrente, relativa all'autenticazione i-esima, perché sarà usata alla prossima autenticazione (la (i+1)-esima) per eseguire il nuovo confronto.

Prof. Montemurro

2 Autenticazione con OTP: Funzione di Hash

Funzione di hash H: funzione unidirezionale, cioè dato w , è facile calcolare $H(w)$, ma, dato $H(w)$, è impossibile calcolare w . Tale funzione è un algoritmo matematico che trasforma dei dati di lunghezza arbitraria in una stringa binaria di lunghezza fissa, chiamata **valore di hash** (poi viene convertita in esadecimale per motivi di leggibilità).

Esempi



Prof. Montemurro

2 Autenticazione con OTP: Funzione di Hash

Gli algoritmi che generano le password OTP vengono inseriti in dispositivi hardware; ogni dispositivo viene associato ad un solo utente, e ogni dispositivo viene inizializzato con un suo **codice-chiave** (il seme w di cui abbiamo parlato prima).

Autenticazione a due fattori 2FA: processo di sicurezza che richiede agli utenti di fornire due diversi fattori di autenticazione per verificare la propria identità (es. password + OTP; carta di credito + PIN; password + impronta digitale). Si usa una combinazione dei tre metodi di autenticazione visti all'inizio (qualcosa di immateriale che l'utente ha, qualcosa di materiale che l'utente possiede, qualcosa di personale dell'utente).

Prof. Montemurro

3a Autenticazione con Firma Digitale

Firma elettronica: espressione priva di qualsiasi valenza tecnico-giuridica che fa riferimento a qualsiasi tecnica (1) utilizzata per l'autenticazione elettronica, e (2) che consente di associare dati ad altri dati (es. la firma ad un documento).

Metodi di autenticazione elettronica ([slide 3](#)):

1. **Metodo di autenticazione basato su qualcosa (di immateriale) che l'utente ha** (password, PIN, ecc.)
2. **Metodo di autenticazione basato su qualcosa (di materiale) che l'utente possiede** (carte fisiche o elettroniche, dispositivi hardware, ecc.)
3. **Metodo di autenticazione basato su qualcosa di personale dell'utente** (dati biometrici come le impronte digitali, la conformazione della retina, ecc.)

Prof. Montemurro

3a Autenticazione con Firma Digitale

Firma digitale: equivalente informatico di una tradizionale firma autografa apposta su carta; si parla di firma digitale perché riguarda un **documento digitale** (o **documento elettronico**, o **documento informatico**), non uno cartaceo.

Caratteristiche della firma digitale

1. **Autenticità:** il destinatario può verificare l'identità del mittente
2. **Integrità:** assicura che il documento non sia stato modificato dopo aver apposto la firma digitale
3. **Non ripudio:** il mittente non può disconoscere un documento da lui firmato, tale documento ha piena validità legale

Prof. Montemurro

3a Autenticazione con Firma Digitale

Principio di funzionamento della firma digitale: basato sulla criptografia asimmetrica; in particolare, il mittente possiede (*):

1. una **chiave privata SK (Secret Key)** la quale non viene condivisa, ed è usata dal mittente per cifrare la firma digitale da apporre sul documento digitale;
2. la corrispondente **chiave pubblica PK (Public Key)** la quale viene condivisa coi destinatari in modo tale che essi possano verificare l'identità del mittente (solo la PK associata alla SK può decifrare i dati cifrati con la SK; una generica PK associata ad un'altra SK non può farlo).

(*) Ricorda che chiave pubblica e privata vengono generate insieme a formare una coppia.

Prof. Montemurro

3a Autenticazione con Firma Digitale

Componenti del kit per apporre la firma digitale



Prof. Montemurro

3a Autenticazione con Firma Digitale

Componenti del kit per apporre la firma digitale

1. **Dispositivo sicuro di generazione delle firme digitali** il quale in genere è una **smart card** nella quale viene memorizzata la chiave privata. Tale dispositivo di firma sicuro viene rilasciato al richiedente da un ente certificatore il quale deve verificare l'identità del richiedente stesso prima di consegnargli (i) la smart card abilitata alla firma digitale, e (ii) il PIN da usare contemporaneamente alla smart card per poter usare la chiave privata
2. **Lettore di smart card**
3. **Software** che, dopo l'attivazione di un account, permette di **apporre la firma digitale** e la **marcatura temporale** sul documento digitale, e che permette di **verificare l'autenticità e l'integrità della firma stessa**
Marcatura temporale: servizio che permette di associare data e ora certe, e legalmente valide ad un documento digitale.

Prof. Montemurro

3a Autenticazione con Firma Digitale

Smart card: dispositivo idoneo per apporre la firma digitale in quanto rispetta i seguenti otto requisiti:

1. apparato elettronico programmabile solo all'origine
2. fa parte del sistema di validazione
3. conserva in modo protetto le chiavi private
4. genera firme digitali al suo interno
5. non è riproducibile
6. è in parte non modificabile
7. l'accesso alla chiave privata è protetto da una procedura di identificazione del titolare (es. inserimento di un PIN)
8. evita di lasciare tracce della chiave privata sul sistema di validazione

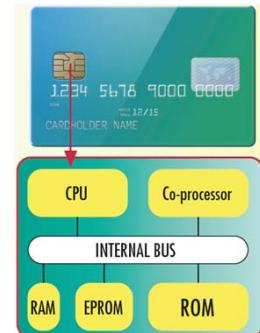
Nota bene: la **Tessera Sanitaria-Carta Nazionale dei Servizi TS-CNS** è una smart card per (i) l'identificazione dell'utente in rete, (ii) la firma digitale, (iii) l'accesso a diversi servizi resi disponibili dalle diverse amministrazioni.

Prof. Montemurro

3a Autenticazione con Firma Digitale

Smart card (continuazione): tessera plastificata, delle dimensioni di una carta di credito, su cui è integrato un **microchip programmabile** con:

1. una memoria (Read Only Memory ROM) che contiene il sistema operativo e i programmi "fissi";
2. una memoria (Programmable Read Only Memory PROM) che contiene il numero seriale della smart card;
3. una terza memoria (Erasable Programmable Read Only Memory EPROM) che contiene i dati del proprietario e i meccanismi di protezione che ne evitano la clonazione.

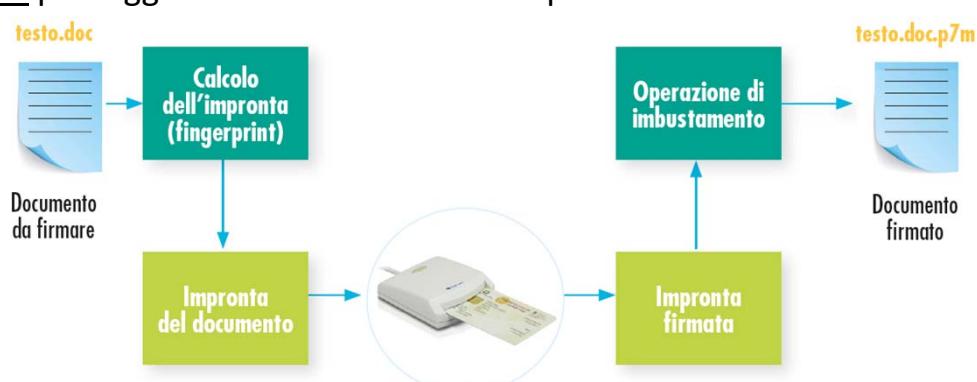


Approfondimento

Prof. Montemurro

3a Autenticazione con Firma Digitale

Formato del documento digitale firmato digitalmente: quando si firma un documento digitale, il file viene incapsulato in una sorta di **busta criptografica**; il risultato è un nuovo file con estensione **.p7m** il quale consente di firmare qualsiasi tipo di file (.doc, .pdf, .xls ecc.). Le pubbliche amministrazioni sono obbligate per legge ad accettare il formato .p7m.



Prof. Montemurro

3b Autenticazione con PEC

Posta Elettronica Certificata PEC: sistema di comunicazione simile alla posta elettronica standard, tale che il mittente sia univocamente identificato.

- Un documento inviato con la PEC assume il medesimo valore legale di una **raccomandata con ricevuta di ritorno**.
- Un'email può essere considerata posta elettronica certificata (PEC) solo se sia la casella del mittente sia quella del destinatario sono caselle di PEC.
- Ogni impresa è obbligata per legge ad avere un indirizzo di PEC iscritto nel Registro delle Imprese INI-PEC, altrimenti l'impresa stessa viene cancellata da tale registro. Ogni ordine (ingegneri, architetti, avvocati, medici) fornisce una PEC gratuitamente.

Prof. Montemurro

3b Autenticazione con PEC

Elenco pubblico dei gestori accreditati: gestori riconosciuti dall'Agenzia per l'Italia Digitale AgID che:

- i. certificano l'identità del mittente e del destinatario;
- ii. certificano l'integrità del messaggio;
- iii. datano con precisione invio e ricezione;

essi sono obbligati a mantenere per 30 mesi (2 anni e mezzo) i dati che riguardano la trasmissione dei messaggi tramite PEC.

Esempio

InfoCert è gestore accreditato di PEC iscritto nell'Elenco Pubblico dei Gestori accreditati, presente nel sito DigitPA, e il suo servizio di Posta Elettronica Certificata è **Legalmail**.

Prof. Montemurro

3b Autenticazione con PEC: Principio di Funzionamento

Premessa: il mittente, dopo aver predisposto il documento da inviare, deve essere identificato dal proprio provider ad esempio tramite user_ID e password.

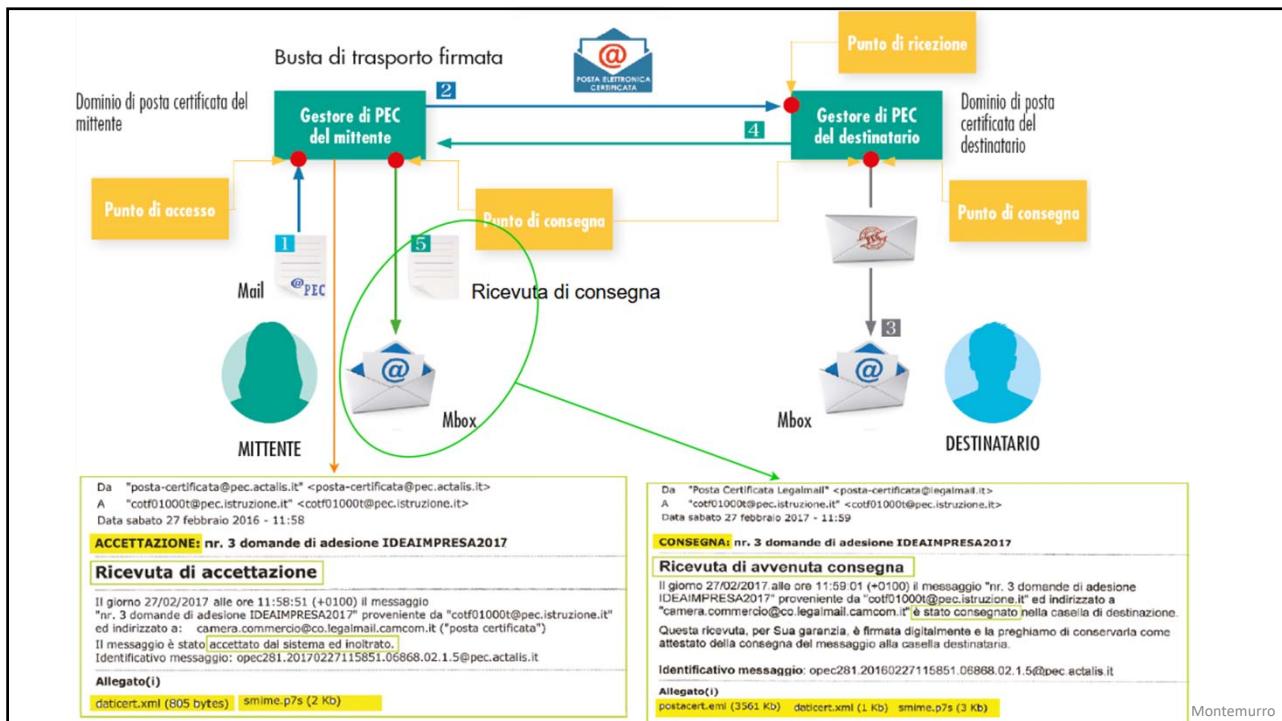
1. Quando il mittente invia un messaggio tramite PEC, egli riceve una **ricevuta di accettazione** del messaggio da parte del suo provider il quale firma tale ricevuta. Essa conferma al mittente la trasmissione del messaggio, e contiene: (i) data e ora dell'invio del messaggio; (ii) il mittente; (iii) il destinatario; (iv) l'oggetto del messaggio.
2. **Integrità del messaggio:** garantita dal provider del mittente il quale crea un nuovo messaggio, detto **busta di trasporto**, costituito (i) dal messaggio originale, e (ii) dai principali dati di spedizione. Tale messaggio viene firmato dal provider del mittente con la sua chiave privata SK; il provider del destinatario, che possiede la chiave pubblica associata alla SK, può usarla per decriptarlo e quindi per constatare la sua integrità.

Prof. Montemurro

3b Autenticazione con PEC: Principio di Funzionamento

3. Il destinatario riceve la busta di trasporto (messaggio + dati spedizione) nella propria casella PEC.
4. Se tutti i controlli fatti danno esito positivo; il provider del destinatario provvede anche a inviare al gestore del mittente la **ricevuta di consegna**, che è un normale messaggio email firmato dal gestore del destinatario che attesta: (i) la ricezione del messaggio da parte del destinatario, (ii) la data e l'ora della consegna, (iii) tutto ciò che è stato consegnato.

Prof. Montemurro



Montemurro

3b Autenticazione con PEC

Vantaggi della PEC

- Risparmio di denaro:** invio di un numero illimitato di comunicazioni senza alcun costo di spedizione; non si devono acquistare buste, lettere, francobolli; non si devono occupare spazi per conservare i documenti cartacei o le ricevute delle poste
- Risparmio di tempo:** si possono spedire documenti dal proprio PC e le ricevute arrivano subito; non si perde tempo a inviare fax o fare code alle poste

Prof. Montemurro



Prof. Montemurro

Fatturazione Elettronica

Classe 5ASI
ITCG Fermi

Prof. Montemurro

Fatturazione Elettronica

Fattura elettronica: equivalente informatico della fattura cartacea. Tale documento elettronico è un file **XML (eXtensible Markup Language)** con estensione **.xml** o **.xml.p7m** (i) che viene firmato digitalmente e (ii) che viene spedito tramite PEC. Tale file contiene sia i dati che i metadati della fattura elettronica.

XML: linguaggio a marcatori usato per scambiare informazioni tra sistemi diversi (es. tra server e applicazioni web, o tra software di aziende differenti)

- L'XML, come l'HTML ha elementi compresi tra un `<tag_di_apertura>` ed un `</tag_di_chiusura>`, ma, a differenza dell'HTML i cui tag sono predefiniti dal linguaggio e , l'XML permette di creare tag personalizzati i quali non servono a mostrare i dati con una certa formattazione come fanno i tag HTML, ma servono solo a racchiudere i dati tra coppie di tag, cioè forniscono informazioni sulla struttura dei dati e sulla loro semantica.

Prof. Montemurro

Fatturazione Elettronica

Vantaggi della fatturazione elettronica per il cittadino e la PA

- 1. Efficiente ed esente da errori**
- 2. Sicura, grazie alla firma digitale ed alla PEC**
- 3. Economica** in quanto non costa nulla realizzarla e non ha costi di affrancatura e di archiviazione fisica
- 4. Veloce** in quanto non richiede tempi di inoltro cartaceo; inoltre, sono rapidi anche i tempi per la sua archiviazione e ricerca
- 5. Tempi di registrazione in contabilità ridotti** in quanto non necessita né di stampa né di distruzione
- 6. Ecosostenibile** dato che anche la archiviazione non richiede la stampa su carta e, di conseguenza, non contribuisce al disboscamento del pianeta (nel solo anno solare 2021 oltre 5 miliardi di file Fattura sono stati scambiati al posto di altrettanti documenti cartacei)

Prof. Montemurro

Fatturazione Elettronica: Struttura della Fattura Elettronica

```
<?xml version="1.0" encoding="UTF-8"?>
<p:FatturaElettronica xsi:schemaLocation="http://.../docs/xsd/fatture/fatturaordinaria_v1.2.xsd "
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:p="http://ivaservizi.agenziaentrate.gov.it/docs/xsd/fatture/v1.2"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#" versione="FPA12">
  + <FatturaElettronicaHeader>
  + <FatturaElettronicaBody>
</p:FatturaElettronica>
```

Tag principale **<p:FatturaElettronica>** contiene due sezioni:

1. **<FatturaElettronicaHeader>** costituita da quattro sottosezioni
2. **<FatturaElettronicaBody>** costituita da tre sottosezioni

Prof. Montemurro

Fatturazione Elettronica: Struttura della Fattura Elettronica



Le informazioni sono riportate in nero.

Prof. Montemurro

Fatturazione Elettronica: Struttura della Fattura Elettronica

Nella sezione **<FatturaElettronicaHeader>**, ed in particolare all'interno della sottosezione **<DatiTrasmissione>** della fattura è necessario inserire il **codice univoco** composto da lettere e numeri, detto **codice ufficio per la fatturazione elettronica**, che l'ente cliente (cioè chi trasmette la fattura) deve comunicare al proprio fornitore:

```

<DatiTrasmissione>
  <CodiceDestinatario>AAAAAAA</CodiceDestinatario>
</DatiTrasmissione>

```

Prof. Montemurro

Fatturazione Elettronica: Struttura della Fattura Elettronica

Regola per nominare il file XML della fattura elettronica

1. **Codice Paese:** codice del paese secondo lo standard ISO (ad esempio "IT" per Italia, "FR" per Francia, ecc.)
2. **Identificativo univoco di chi trasmette la fattura:** ad esempio, in Italia potrebbe essere il codice fiscale o la partita IVA del soggetto trasmittente
3. **Progressivo univoco del file:** numero o codice che **distingue questo file** da tutti gli altri, per evitare che ci siano file con lo stesso nome. Questo numero cambia ogni volta che invii una nuova fattura (esempio: "0001", "0002", "0003", ecc.)

Prof. Montemurro

Fatturazione Elettronica

Obiettivo dell'invio della fatturazione elettronica alla PA: fornitore della PA chiede alla PA stessa di farsi pagare per i prodotti e/o servizi forniti.

Invio della fattura elettronica alla PA: il postino che porta la fattura elettronica, firmata digitalmente, **dal** fornitore (o ente cliente) **alla** PA competente è il

Sistema di Interscambio SdI. Tale sistema:

1. riceve la fattura elettronica digitalmente firmata dal fornitore;
2. controlla l'autenticità e l'integrità del file inviato alla PA, se questi requisiti sono soddisfatti, appone sul file della fattura elettronica la marcatura temporale;
3. inoltra alla PA competente la fattura elettronica.

Prof. Montemurro



Prof. Montemurro

Approvvigionamento Elettronico e Mercato Elettronico della PA

Classe 5ASI
ITCG Fermi

Prof. Montemurro

Approvvigionamento Elettronico

Approvvigionamento elettronico (o **e-procurement**): applicazione che permette ad un'azienda o ad un governo di acquistare in modo automatico (*) beni e servizi di cui ha bisogno per la propria attività.

(*) Senza bisogno di interventi manuali continui; il sistema informatico può gestire in modo automatico:

1. l'identificazione dei fornitori;
2. la richiesta di offerte;
3. la gestione degli ordini;
4. in alcuni casi anche la fatturazione e il pagamento.

Prof. Montemurro

Approvvigionamento Elettronico

Modello di business: non esiste una definizione univocamente accettata; nella maggior parte delle definizioni ci sono tre elementi in comune:

1. la **proposta di valore**, cioè i beni e/o i servizi che l'azienda offre ai suoi clienti;
2. la **creazione di valore** (elemento più importante), cioè i metodi con cui l'azienda soddisfa i bisogni dei clienti e fa crescere sé stessa;
3. la **cattura del valore**, cioè la conversione dei pagamenti dei clienti in profitti.

Prof. Montemurro

Approvvigionamento Elettronico

Tipi di modelli di business (questi riguardano sia il commercio elettronico che non elettronico)

	Government	Business	Consumer	Employee
Government	G2G	G2B	G2C	G2E
Business	B2G	B2B	B2C	B2E
Consumer	C2G	C2B	C2C	

Tre principali modelli di business

1. **B2B (Business-to-Business)**, cioè commercio tra aziende
2. **B2C (Business-to-Customer)**, cioè commercio tra aziende e privati cittadini
3. **G2B (Government-to-Business)**, cioè commercio tra istituzioni pubblico e aziende; noi ci soffermeremo su questo

Prof. Montemurro

Approvvigionamento Elettronico della PA

Approvvigionamento elettronico della PA: sistema di approvvigionamento elettronico della PA, si tratta di una piattaforma online, realizzata dal **MEF** (Ministero dell'Economia e delle Finanze) tramite la società **Consip S.P.A.** la quale gestisce le procedure di acquisto di beni e/o servizi della PA stessa.



- Le amministrazioni statali centrali e periferiche sono obbligate per legge ad eseguire le procedure di acquisto di beni e/o servizi tramite la piattaforma.
- (1) Istituti e scuole di ogni ordine e grado, (2) istituzioni educative, e (3) istituzioni universitarie non sono obbligate ad usare la piattaforma.

Prof. Montemurro

Approvvigionamento Elettronico della PA

Strumenti di acquisto/vendita con cui (i) le imprese offrono i loro beni e/o servizi alle PA, e con cui (ii) le PA effettuano acquisti sono quattro:

1. Convenzioni
2. Accordi quadro
3. Sistema Dinamico di Acquisizione (SDA)
4. Mercato elettronico della PA (MePA)

Noi ci concentreremo soprattutto sul MePA.

Prof. Montemurro

Approvvigionamento Elettronico della PA

Soglia comunitaria: soglia stabilita dall'unione europea, compresa tra € 130.000 e 200.000 a seconda della tipologia di beni e servizi che la PA intende acquistare.

1. Strumenti per acquisti di beni e servizi di importo sopra e sotto soglia: convenzioni, accordi quadro, Sistema Dinamico di Acquisizione (SDA)
2. Strumenti per acquisti di beni e servizi di importo sotto soglia: Mercato elettronico della PA (MePA); unico strumento che non consente acquisti sopra soglia

	CONVENZIONE	ACCORDO QUADRO	MERCATO ELETTRONICO	SISTEMA DINAMICO
SOGLIA COMUNITARIA	sopra e sotto soglia	sopra e sotto soglia	sotto soglia	sopra e sotto soglia

Prof. Montemurro

Approvvigionamento Elettronico della PA

Catalogo del fornitore: può contenere prodotti, servizi, prezzi, quantità, specifiche tecniche.

	CONVENZIONE	ACCORDO QUADRO	MERCATO ELETTRONICO	SISTEMA DINAMICO
PRESENZA DEL CATALOGO	sì	sì	sì	no

Prof. Montemurro

Approvvigionamento Elettronico della PA

Canali di acquisto

- Ordini Diretti d'Acquisto (ODA):** la PA acquista beni o servizi direttamente dal catalogo del fornitore senza necessità di bandire una gara d'appalto.
- Richieste Di Offerta (RDO) (o negoziazioni):** la PA individua e descrive i beni e/o servizi che intende acquistare, ed invita i fornitori a presentare le specifiche offerte che saranno oggetto di confronto concorrenziale (chi offre gli stessi beni/servizi a prezzo più basso vince).



Prof. Montemurro

Approvvigionamento Elettronico della PA

Modalità di accesso delle imprese agli strumenti di acquisto/vendita della PA

1. Partecipazione alla gara d'appalto

Gara d'appalto: procedura amministrativa con cui una stazione appaltante (es. ente pubblico) seleziona, tra più operatori economici, quello che presenta l'offerta migliore per l'esecuzione di lavori, la fornitura di beni o la prestazione di servizi (chi offre gli stessi beni/servizi a prezzo più basso vince).

2. Abilitazione al bando: accesso mirato a specifiche categorie del bando, non all'intero bando

3. Ammissione al bando: autorizzazione a partecipare all'intero bando



1 Approvvigionamento Elettronico della PA: Convenzioni

Convenzioni: contratti che le PA possono usare per l'acquisto o il noleggio di beni e servizi. Esse hanno (1) una **scadenza temporale** e (2) prevedono un **quantitativo** (es. max 1000 stampanti) o un **importo massimo (massimale)** (es. max 2 milioni di euro) che il fornitore si impegna a garantire nell'ambito del contratto.

CONVENZIONE	
SOGLIA COMUNITARIA	sopra e sotto soglia
MODALITA' DI ACQUISTO	ordine diretto
POSSIBILITA' DI PERSONALIZZAZIONE	bassa
PRESENZA DEL CATALOGO	sì
ACCESSO ALLE IMPRESE	partecipazione alla gara

Prof. Montemurro

1 Approvvigionamento Elettronico della PA: Convenzioni

Funzionamento delle convenzioni



- Consip, essendo la centrale di committenza per le PA, sceglie il vincitore della gara d'appalto, e gli conferisce il diritto ed il dovere di eseguire il contratto.
- Centrale di committenza:** è una stazione appaltante o un ente concedente che fornisce attività di centralizzazione delle committenze in favore di altre stazioni appaltanti o enti concedenti, fornendo, se necessario, supporto all'attività di committenza.

Prof. Montemurro

2 Approvvigionamento Elettronico della PA: Accordi Quadro

Accordi Quadro (AQ): sono contratti quadro che le PA possono usare per l'acquisto o il noleggio di beni e servizi. Essi hanno (1) una **scadenza temporale** e (2) prevedono un **quantitativo** (es. max 1000 stampanti) o un **importo massimo (massimale)** (es. max 2 milioni di euro) che il fornitore si impegna a garantire nell'ambito del contratto.

Contratto quadro: incarico, ovvero una sorta di mandato, conferito all'intermediario finanziario, affinché costui esegua gli incarichi di volta in volta conferitigli dall'investitore (sono incarichi che saranno conferiti in futuro).

Prof. Montemurro

2 Approvvigionamento Elettronico della PA: Accordi Quadro

ACCORDO QUADRO

SOGLIA COMUNITARIA	sopra e sotto soglia
MODALITA' DI ACQUISTO	ordine diretto o negoziazione
POSSIBILITA' DI PERSONALIZZAZIONE	medio / alta
PRESENZA DEL CATALOGO	sì
ACCESSO ALLE IMPRESE	partecipazione alla gara

Prof. Montemurro

2 Approvvigionamento Elettronico della PA: Accordi Quadro

Funzionamento degli accordi quadro



- Consip, essendo la centrale di committenza per le PA, sceglie uno o più vincitori della gara d'appalto, e gli conferisce il diritto ed il dovere di eseguire il contratto.

Prof. Montemurro

3 Approvvigionamento Elettronico della PA: SDA

Sistema Dinamico di Acquisizione (SDA): tipo di mercato digitale della PA per l'acquisto di beni e servizi.

SISTEMA DINAMICO	
SOGLIA COMUNITARIA	sopra e sotto soglia
MODALITA' DI ACQUISTO	negoziazione
POSSIBILITA' DI PERSONALIZZAZIONE	alta
PRESENZA DEL CATALOGO	no
ACCESSO ALLE IMPRESE	ammissione al Bando

Prof. Montemurro

3 Approvvigionamento Elettronico della PA: SDA

Funzionamento del sistema dinamico di acquisizione



- Consip, essendo la centrale di committenza per le PA, sceglie il vincitore della gara d'appalto, e gli conferisce il diritto ed il dovere di eseguire il contratto.

Prof. Montemurro

4 Approvvigionamento Elettronico della PA: MePA

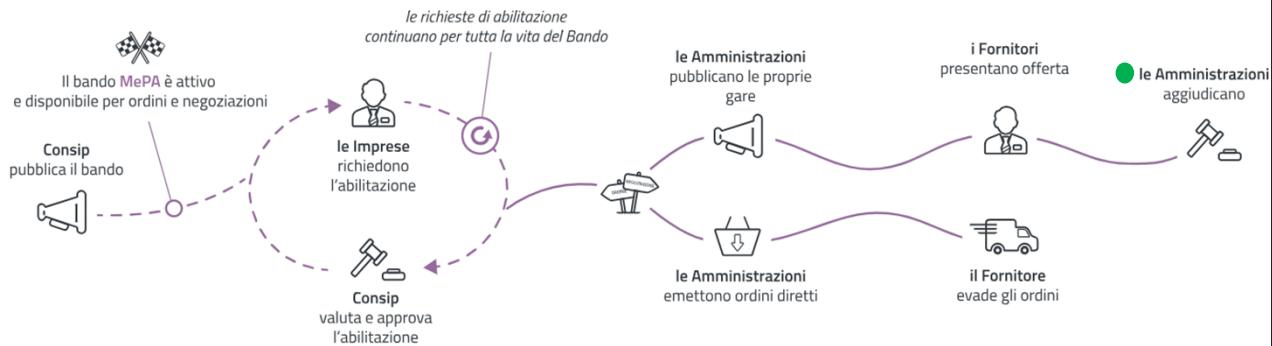
Mercato elettronico della PA (MePA): tipo di mercato digitale della PA per gli acquisti, sotto soglia comunitaria, di beni, servizi, e lavori di manutenzione.

MERCATO ELETTRONICO	
SOGLIA COMUNITARIA	sotto soglia
MODALITA' DI ACQUISTO	ordine diretto o negoziazione
POSSIBILITA' DI PERSONALIZZAZIONE	alta
PRESENZA DEL CATALOGO	sì → sì per i beni sì per alcune tipologie di servizi no per i lavori
ACCESSO ALLE IMPRESE	abilitazione al Bando

Prof. Montemurro

4 Approvvigionamento Elettronico della PA: MePA

Funzionamento del mercato elettronico della PA



- Consip, essendo la centrale di committenza per le PA, sceglie il vincitore della gara d'appalto, e gli conferisce il diritto ed il dovere di eseguire il contratto.

Prof. Montemurro

4 Approvvigionamento Elettronico della PA: MePA

Operatori economici ammessi al MePA: operatori economici che possono partecipare alle procedure di affidamento dei contratti pubblici in conformità (i) al d. lgs. n. 163/2006 e (ii) alla normativa comunitaria in materia.

Tipi di operatori economici

1. Singolo operatore economico (società o impresa individuale anche artigiana)
2. Consorzio fra società cooperative di produzione e lavoro
3. Rete d'impresa con soggettività giuridica
4. Rete di imprese con soggettività giuridica operante con propria organizzazione d'impresa

Prof. Montemurro

4 Approvvigionamento Elettronico della PA: MePA

Procedura di accesso al MePA: composta da due fasi distinte e successive da eseguire su <https://www.acquistinretepa.it>

1. **Registrazione base:** inserimento di dati per dire "chi sei"
2. **Abilitazione** (accesso mirato ad alcune categorie dei bandi): inserimento di dati per specificare "che fai"; l'abilitazione si richiede tramite i bandi di abilitazione.

Bando di abilitazione: bando sempre aperto nel tempo per tutti gli operatori economici i quali quindi si possono iscrivere in qualunque istante.

Prof. Montemurro

4 Approvvigionamento Elettronico della PA: MePA

Forniture classificate in **aree merceologiche**; cliccando si accede ai bandi aperti

 ALIMENTI, RISTORAZIONE E BUONI PASTO	 ARREDI, COMPLEMENTI ED ELETTRODOMESTICI	 ATTREZZATURE, INDUMENTI E DPI	 CANCELLERIA, CARTA E CONSUMABILI
 EDITORIA, EVENTI E COMUNICAZIONE	 ENERGIA, CARBURANTI E LUBRIFICANTI	 FERRAMENTA, IDRAULICA, EDILIZIA E MATERIALE ELETTRICO	 GESTIONE DEGLI IMMOBILI
 INFORMATICA, ELETTRONICA, TELECOMUNICAZIONI E MACCHINE PER L'UFFICIO	 LAVORI DI MANUTENZIONE E OPERE PUBBLICHE	 MACCHINARI, STRUTTURE LOGISTICHE E ABITATIVE	 MONOUSO, PULIZIE E IGIGNE PERSONALE
 RIFIUTI	 SANITÀ, RICERCA E WELFARE	 SERVIZI PER IL FUNZIONAMENTO DELLE P.A.	 STRADE, VERDE PUBBLICO E GESTIONE DEL TERRITORIO
 VEICOLI, MOBILITÀ E TRASPORTI			

Prof. Montemurro

4 Approvvigionamento Elettronico della PA: MePA

Area merceologica

Sanità, Ricerca e Welfare

Apparecchiature elettromedicali

Accessori e materiali specifici

Assistenza, manutenzione e riparazione

Cardiologia

Defibrillatori

Ectomografia

Endoscopia

Medicina generale

Monitoraggio glicemia

Monitoraggio parametri vitali

Neonatologia

[VEDI TUTTI](#)

Arredi da laboratorio, sanitari e post mortem

Arredi per ambulatori e medicherie

Arredi per degenza

Arredi per sale e blocchi operatori

Arredi per sale parto

Arredi post-mortem

Arredi tecnici

Assistenza, manutenzione e riparazione

Barelle

Carrelli

Letti da ospedale

[VEDI TUTTI](#)

Dispositivi medici

Aghi

Apparecchiature e dispositivi per nutrizione entrale

Aspirazione chirurgica

Assistenza, manutenzione e riparazione

Assorbenti

Ausili per incontinenza

Ausili tecnici per persone disabili

Circuiti respiratori e cateteri mount

Coperture chirurgiche e altri prodotti per sala operatoria e ambulatorio chirurgico

Coperture sanitarie non chirurgiche

[VEDI TUTTI](#)

Monitoraggio ambientale e sanitario

Monitoraggi e verifiche per le strutture sanitarie e di ricerca

Monitoraggio ambientale - Campi elettromagnetici

Monitoraggio ambientale - Qualità dell'aria

Servizi sociali e di welfare

Accoglienza ed accompagnamento alla residenzialità

Accompagnamento all'integrazione sociale per comunità emarginate

Assistenziali, educativi, segretariato, supporto al lavoro

4 Approvvigionamento Elettronico della PA: MePA



Date Iniziativa



Requisiti per l'ammissione



Criterio di aggiudicazione (AS)



Acquisti verdi

SDA Arredi - Novità categorie merceologiche

Si comunica che in data odierna sono state pubblicate alcune novità relative al Bando del ...

BANDI SIMILI

Dispositivi impiantabili per resincronizzazione cardiaca (CRT)

2

Data di pubblicazione del bando/avviso

18/06/2020

Data di attivazione

18/06/2020

Data di scadenza

18/06/2022

Infine viene richiesto di inserire i documenti necessari per l'ammissione al bando.

Approvvigionamento Elettronico della PA: Sinottico

	CONVENZIONE	ACCORDO QUADRO	MERCATO ELETTRONICO	SISTEMA DINAMICO
SOGLIA COMUNITARIA	sopra e sotto soglia	sopra e sotto soglia	sotto soglia	sopra e sotto soglia
MODALITA' DI ACQUISTO	ordine diretto	ordine diretto o negoziazione	ordine diretto o negoziazione	negoziante
POSSIBILITA' DI PERSONALIZZAZIONE	bassa	medio / alta	alta	alta
PRESENZA DEL CATALOGO	sì	sì	sì	no
ACCESSO ALLE IMPRESE	partecipazione alla gara	partecipazione alla gara	abilitazione al Bando	ammissione al Bando



Identificazione Digitale, Spid, PagoPA

Classe 5ASI
ITCG Fermi

Prof. Montemurro

Metodi di Autenticazione Elettronica (o di Identificazione Digitale)

1. **Metodo di autenticazione basato su qualcosa (di immateriale) che l'utente ha** (password, PIN, ecc.)
2. **Metodo di autenticazione basato su qualcosa (di materiale) che l'utente possiede** (carte fisiche o elettroniche, dispositivi hardware, ecc.)
3. **Metodo di autenticazione basato su qualcosa di personale dell'utente** (dati biometrici come le impronte digitali, la conformazione della retina, ecc.)

Prof. Montemurro

SPID

Sistema Pubblico di Identità Digitale (SPID): sistema di identificazione digitale, realizzato dalla PA, che garantisce a tutti i cittadini (anche minorenni) ed alle imprese un accesso sicuro ai dati ed ai servizi digitali erogati (1) dalle PA a livello nazionale e locale, e (2) dai fornitori di servizi privati aderenti.



Oggi i servizi in rete delle PA sono accessibili solo con SPID, CIE, e TS-CNS.

Prof. Montemurro

SPID: Livelli di Sicurezza

Livelli di sicurezza delle credenziali SPID (tre livelli)

- Primo livello di sicurezza (o credenziali di livello 1):** username e password; è un'autenticazione a singolo fattore
- Secondo livello di sicurezza (o credenziali di livello 2):** username e password, OTP; è un'autenticazione a due fattori
- Terzo livello di sicurezza (o credenziali di livello 3):** username e password, supporto hardware per l'identificazione tipo una smart card (es. TS-CNS), ed eventualmente un PIN o riconoscimento biometrico; è un'autenticazione a due/tre fattori

Prof. Montemurro

SPID: Come Ottenerlo

Requisiti per ottenere lo SPID

1. Indirizzo email
2. Numero di cellulare
3. Documento di identità valido (carta d'identità, passaporto, patente, permesso di soggiorno)
4. Tessera Sanitaria - Carta Nazionale dei Servizi (TS-CNS)

Lo SPID può essere richiesto:

- i. dai cittadini maggiorenni;
- ii. dai cittadini minorenni tramite un loro genitore che deve rivolgersi al proprio gestore di identità digitale con credenziali di livello 2.

Prof. Montemurro

SPID: Come Ottenerlo

Gestori di identità digitale: soggetti autorizzati dalla PA a rilasciare lo SPID (es. Poste Italiane che lo rilascia gratuitamente, Aruba, Tim ecc.).

Procedura per ottenere lo SPID

1. Compilazione di un modulo coi dati anagrafici
2. Verifica dell'email e del cellulare del richiedente da parte dal gestore di identità digitale

Nota bene: un cittadino può richiedere più di uno SPID, anche con diversi livelli di sicurezza; per farlo, basta che si rivolga a diversi gestori di identità digitale (es. un cittadino può avere uno SPID con Poste Italiane, un altro SPID con ARUBA ecc.).

Prof. Montemurro

Sistema di Pagamento PagoPA

PagoPA: sistema di pagamenti elettronici realizzato dalla PA, per eseguire pagamenti verso (1) le PA, ed (2) i gestori di pubblici servizi tramite i **Prestatori di Servizi di Pagamento PSP** aderenti (es. banche, Poste, istituti di pagamento, ogni altro soggetto abilitato ad eseguire servizi di pagamento a cittadini ed imprese). Tutte le PA e le società a controllo pubblico sono obbligate per legge ad aderire al sistema PagoPA.

Modalità di pagamento previste dal sistema PagoPA

1. Sul sito di un ente a cui si deve versare una quota
2. Sull'eventuale app di un ente a cui si deve versare una quota
3. Canali fisici e online dei PSP (es. sportelli bancari, uffici postali, home banking, sportelli ATM, punti vendita SISAL, Lottomatica, Banca 5)

Prof. Montemurro

Sistema di Pagamento PagoPA

Caratteristiche del sistema PagoPA

1. Non oneroso: no costi aggiuntivi a carico della PA o del cittadino legati all'utilizzo del sistema PagoPA stesso
2. Modalità elettroniche di pagamento standardizzate (stesso modo di pagare a prescindere che si vada in banca, alle Poste ecc.)
3. PA e PSP possono ricevere qualsiasi tipo di pagamento (tributi, tasse, utenze, rette, quote associative, bolli)
4. Incassi automatizzati: incassi ricevuti, registrati, e riconciliati in modo automatico, senza alcun intervento manuale per cui i costi di gestione degli incassi sono ridotti
5. Costi di commissione applicati dai PSP messi in evidenza

Prof. Montemurro



Prof. Montemurro