

# **Introduzione allo sviluppo Web**

Studium Generale, a.a. 2013-2014, II semestre

# Contatto

- ▶ **Daniel Graziotin**
- ▶ **daniel.graziotin@unibz.it**
- ▶ **[http://ineed.coffee/teaching/  
introduzione-allo-sviluppo-web](http://ineed.coffee/teaching/introduzione-allo-sviluppo-web)**

# Esame

## ▶ **Esame ufficiale**

- Giovedì 26 Giugno, 17-19 E231

## ▶ **Invio progetti:**

- Venerdì 20 Giugno, mezzanotte
- [daniel.graziotin@unibz.it](mailto:daniel.graziotin@unibz.it)

# Esame

## ▶ **Formato**

## ▶ **Archivio (es: .zip) del sito**

- Estratto nella root / del mio webserver
- Autocontenuto (no file esterni)
- NON date per scontato che si chiami localhost

# PHP

## I cicli

### **Ripetere operazioni in maniera ripetitiva**

▶ **“Da X a Y volte, fai questa cosa”**

- Ciclo for

▶ **“Fino a quando una condizione è vera, fai questa cosa”**

- Ciclo while

# PHP

## Il ciclo for

- ▶ **“Da X a Y volte, fai questa cosa”**
- ▶ **“Da X a Y volte”**
- ▶ **Fai Questa Cosa**

# PHP

## Il ciclo for

### ▶ **“Da X”**

- `$contatore = X;`

### ▶ **“a Y volte”**

- `$contatore <= NUMERO_VOLTE;`
- `$contatore++;`

### ▶ **Fai Questa Cosa**

# PHP

## Il ciclo for

### ▶ “Da 1 a 10 volte”

- `$contatore = 1;` # Da 1
- `$contatore <= 10;` # a 10 volte
- `$contatore++;` # (nuovo giro)

### ▶ **Fai Questa Cosa**



# PHP

## Il ciclo for

### ▶ “Da 1 a 10 volte”

- `$i = 1;` # Da 1
- `$i <= 10;` # a 10 volte
- `$i++;` # (nuovo giro)

### ▶ Fai Questa Cosa

{

**# istruzioni da ripetere**

}

# PHP

## Il ciclo for

- ▶ **“Da 1 a 10 volte, scrivi ‘ciao’ seguito da un invio”**

```
for ($i = 1; $i <= 10; $i++) {  
    echo "ciao";  
    echo "<br/>";  
}
```

# PHP

## Il ciclo for

- ▶ **“Da 1 a 20 volte, scrivi ‘ciao’ seguito da un invio”**

```
for ($i = 1; $i <= 20; $i++) {  
    echo "ciao";  
    echo "<br/>";  
}
```

# PHP

## Il ciclo for

- ▶ **“Da 1 a 20 volte, scrivi ‘ciao’ seguito dal numero di volta seguito da un invio”**

```
for ($i = 1; $i <= 20; $i++) {  
    echo "ciao ";  
    echo $i;  
    echo "<br/>";  
}
```

# PHP

## Il ciclo while

- ▶ **“Fino a quando una condizione è vera, fai questa cosa”**
- ▶ **Fino a quando condizione vera**
- ▶ **Fai questa cosa**

# PHP

## Il ciclo while

- ▶ **“Fino a quando una condizione è vera, fai questa cosa”**
- ▶ **Fino a quando condizione vera**  
`$contatore > 0` (`> 10`, `< 1`, `!= 3`, ....., va bene tutto)
- ▶ **Fai questa cosa**  

```
{  
    # operazioni da ripetere  
    $contatore++;      # (o --)  
}
```

# PHP

## Il ciclo while

- ▶ **“Fino a quando una condizione è vera, fai questa cosa”**

- ▶ **Fino a quando condizione vera**

```
while ($contatore > 0)
```

- ▶ **Fai questa cosa**

```
{
```

```
    # operazioni da ripetere
```

```
    $contatore++;      # (o --)
```

```
}
```

# PHP

## Il ciclo for

- ▶ **“Fino a quando il contatore è maggiore di 0, scrivi ‘ciao’ seguito da un invio”**

```
$i = 10;
```

```
while ($i > 0) {
```

```
    echo "ciao";
```

```
    echo "<br/>";
```

```
    $i--;
```

```
}
```



# PHP

## Il ciclo for

- ▶ **“Fino a quando il contatore è maggiore di 0, scrivi ‘ciao’ seguito da un invio”**

```
$i = 10;
```

```
while ($i > 0) {
```

```
    echo "ciao";
```

```
    echo "<br/>";
```

```
    $i--;                # perché non $i++?
```

```
}
```

# PHP

## Il ciclo foreach

- ▶ **Ciclo speciale per array: foreach**
- ▶ **Per ogni elemento di un array, fai qualcosa**

```
foreach ($nome_array as $elemento){
```

```
    # fai qualcosa
```

```
    # per ogni ciclo, $elemento prende il valore
```

```
    # dell'elemento attuale dell'array
```

```
}
```

# PHP

## Il ciclo foreach

- ▶ **Ciclo speciale per array: foreach**
- ▶ **Per ogni elemento di un array, fai qualcosa**

**ritorno.php**

```
foreach($dati as $dato){  
    echo "Dato spedito: ".$dato;  
    echo "<br/>";  
}
```

# PHP

## Il ciclo foreach

- ▶ **Ciclo speciale per array: foreach**
- ▶ **Per ogni elemento di un array, fai qualcosa**

**ritorno.php**

```
foreach($dati as $chiave => $valore){  
    echo $chiave . " => " . $valore;  
    echo "<br/>";  
}
```

# PHP

## Includere file

- ▶ **PHP permette di includere file PHP dentro a file PHP**
  - Una specie di copia-incolla automatico
- ▶ **Utile per evitare di riscrivere codice PHP**
  - Anche HTML!

# PHP

Includere file

- ▶ **Caso base: HTML**
- ▶ **Osserviamo index.php, andata.php, ritorno.php**
- ▶ **Cosa hanno in comune?**

# PHP

## Includere file

- ▶ **Cosa hanno in comune index.php, andata.php, ritorno.php?**
  - Tutto `<head>` **tranne** `<title>`
  - Tutto `<div class="navigation">` (Il menu)
  - Tutto `<div class="side-bar">` (la sidebar)
  - Tutto `<div class="footer">` (i footers)

# PHP

## Includere file

- ▶ **Cosa hanno in comune index.php, andata.php, ritorno.php?**
  - Tutto `<head>` **tranne** `<title>`
  - **Tutto** `<div class="navigation">` (Il menu)
  - Tutto `<div class="side-bar">` (la sidebar)
  - Tutto `<div class="footer">` (i footers)



# PHP

## Includere file

### ▶ Il menu

```
<div class="navigation">
```

```
  <ul class="menu">
```

```
    <li><a class="active" href="#">chi siamo</a></li>
```

```
    <li><a href="#">cosa facciamo</a></li>
```

```
    <li><a href="#">contatti</a></li>
```

```
    <li><a href="#">links</a></li>
```

```
  </ul>
```

```
</div>
```

# PHP

## Includere file

- ▶ Mettiamolo in un file chiamato **navigation.php**

```
<div class="navigation">
```

```
  <ul class="menu">
```

```
    <li><a class="active" href="#">chi siamo</a></li>
```

```
    <li><a href="#">cosa facciamo</a></li>
```

```
    <li><a href="#">contatti</a></li>
```

```
    <li><a href="#">links</a></li>
```

```
  </ul>
```

```
</div>
```

# PHP

## Includere file

- ▶ **navigation.php non contiene un documento HTML valido**
- ▶ **Solo div.navigation e i suoi contenuti**
- ▶ **Va bene così**
  - Il suo contenuto viene copiato e incollato negli altri file

# PHP

## Includere file

- ▶ **Apriamo index.php**
- ▶ **Cancelliamo tutto div.navigation e suoi contenuti**
- ▶ **Inseriamo il seguente al posto di div.navigation**
  - Quindi prima di div.main-content

```
<?php
```

```
    require_once("navigation.php");
```

```
?>
```

# PHP

## Includere file

- ▶ **Cosa succede quando accediamo a <http://localhost/index.php>**
- ▶ **Il menu viene incluso**

# PHP

## Includere file

- ▶ **Apriamo andata.php e ritorno.php**
- ▶ **Cancelliamo tutto div.navigation e suoi contenuti**
- ▶ **Inseriamo il seguente al posto di div.navigation**
  - Quindi prima di div.main-content

```
<?php
```

```
    require_once("navigation.php");
```

```
?>
```

# PHP

## Includere file

- ▶ **In con `require_once("file.php")`, il contenuto di `file.php` viene incluso (copiato e incollato) dentro al file che chiama `require_once`**
- ▶ **Comodo**
  - Scrivere un menu una sola volta

# PHP

## Includere file

- ▶ **Cosa hanno in comune index.php, andata.php, ritorno.php?**
  - Tutto `<head>` **tranne** `<title>`
  - Tutto `<div class="navigation">` (Il menu)
  - Tutto `<div class="side-bar">` (la sidebar)
  - Tutto `<div class="footer">` (i footers)



# PHP

## Includere file

- ▶ **Cosa hanno in comune index.php, andata.php, ritorno.php?**
  - Tutto `<head>` tranne `<title>`
  - Tutto `<div class="navigation">` (Il menu)
  - Tutto `<div class="side-bar">` (la sidebar)
  - Tutto `<div class="footer">` (i footers)

# PHP

## Includere file

- ▶ **Al momento, il contenuto di `<head>` è in comune eccetto il title.**

**`<head>`**

**`<meta charset="utf-8">`**

**`<title>basic page template</title>`**

**`<link rel="stylesheet" type="text/css" href="/style.css">`**

**`</head>`**

- ▶ **Come fare per raggruppare questo codice?**

# PHP

## Includere file

- ▶ File **meta.php** con dentro:

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

```
<link rel="stylesheet" type="text/css" href="/style.css">
```

# PHP

## Includere file

- ▶ File **meta.php** con dentro:

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

```
<link rel="stylesheet" type="text/css" href="/style.css">
```

- ▶ Incluso con

```
<head>
```

```
<?php  
    require_once("meta.php");  
?>
```

```
<title>basic page template</title>
```

```
</head>
```

# PHP

Cosa abbiamo visto

## ▶ Sintassi base

- `<?php e ?>`
- Istruzioni
- echo (comprese concatenazioni)
- Commenti `//` `#` `/* */`

# PHP

## Cosa abbiamo visto

### ▶ Variabili

- Modello della “scatola di memoria”
- Assegnazioni di valore `$nome = "Daniel";`
- Tipi di variabile (stringhe, numeri, booleani)
- Operazioni matematiche

# PHP

## Cosa abbiamo visto

### ▶ Routine

- Procedure (senza return)
- Funzioni (con return)
- Parametri (o argomenti)

```
function somma($num1, $num2){}
```

# PHP

## Cosa abbiamo visto

### ▶ **Array**

- Modello della collezione
- Semplici `$studenti[0] = "Naomi";`
- Associativi `$capitali["italia"] = "Roma";`
- Chiavi di array
- `print_r()`
- Modificare array (cancellare, contare, ordinare)



# PHP

Cosa abbiamo visto

## ▶ **Form (HTML)**

- GET e POST
  - A livello di protocollo HTTP
- Input e textarea
- `$_GET` e `$_POST`

# PHP

Cosa abbiamo visto

## ▶ **Strutture di controllo**

- If-else
- Booleani (operazioni e valori)

## ▶ **Cookies**

- A livello di protocollo HTTP
- `setcookie()`
- `$_COOKIE`

# PHP

Cosa abbiamo visto

## ▶ **Cicli**

- For
- While
- Foreach

## ▶ **Inclusione dei file**

- `require_once`

# PHP

Cosa abbiamo visto

▶ **Il resto?**

# I Dogmi

## HTML

- ▶ **Un documento HTML è struttura e contenuto**
  - markdown
  - mai grafica o dinamicità
  - se trovate documenti che vi istruiscono di mettere stile nei tag, ignorateli

# I Dogmi

## HTML

### ▶ I tag sono sempre dentro ai simboli < e >

- Si aprono e chiudono SEMPRE (per noi)
  - <tag> e </tag>
  - <tag />
  - Eccezione: <!DOCTYPE >
  - <a href="<http://ineed.coffee>">link</a>
  - Non <a href="<http://ineed.coffee>">link</a href="<http://ineed.coffee>">

# I Dogmi

## HTML

### ▶ **Gli attributi dei tag sono SEMPRE in forma nome="valore"**

- Sempre dopo il nome del tag
  - `<a href="http://ineed.coffee" id="miosito">` non `<id="miosito" href="http://ineed.coffee" a>`
- Il valore va sempre tra apici " "
- Attenzione al copia e incolla
  - Corretto: " e '   
▪
  - Scorretto: ” e ‘

# I Dogmi

## HTML

- ▶ **Un documento HTML ha tre tag fondamentali**
  - `<html>` racchiude il documento intero eccetto il doctype
    - `<head>` contiene tag di "istruzioni"
    - `<body>` contiene tag per la fruizione del documento stesso



# I Dogmi

## HTML

### ▶ Un documento HTML ha tre tag fondamentali

- `<html>` racchiude il documento intero eccetto il doctype
  - `<head>` contiene tag di "istruzioni"
    - I tag per il charset, per includere CSS (o JavaScript) o metadati
  - `<body>` contiene tag per la fruizione del documento stesso
    - Tutti gli altri tag (p, a, div, span, table, h1 ecc)

# I Dogmi

## CSS

- ▶ **Un foglio di stile CSS è grafica, presentazione**
  - I documenti HTML includono i fogli di stile CSS
  - Sono regole per il browser

# I Dogmi

## CSS

- ▶ **La sintassi base dei fogli di stile è:**

**selettore {**

**proprietà: valore;**

**proprietà: valore;**

**}**

- ▶ **Ogni dichiarazione CSS termina con un punto e virgola**

# I Dogmi

## CSS

- ▶ Per stilare certi gruppi di tag si usano le classi CSS

```
tag.nomeClasse {  
    proprietà: valore;  
    proprietà: valore;  
}
```

- ▶ In HTML c'è un corrispondente

```
<tag class="nomeClasse">
```

**Tante classi con lo stesso nome** per documento HTML

# I Dogmi

## CSS

- ▶ Per stilare dei tag in particolare di usano gli id

```
tag#nomeld{
```

```
    proprietà: valore;
```

```
    proprietà: valore;
```

```
}
```

- In HTML c'è un corrispondente

```
<tag id="nomeld">
```

- ▶ **Un solo ID** per documento HTML

# I Dogmi

## PHP

- ▶ **Un file PHP è programmazione lato server**
  - In questo corso: PHP si mischia con HTML
    - Si può evitare

# I Dogmi

## PHP

- ▶ **Il codice php sta sempre tra i tag `<?php e ?>`**
  - Mai fuori, “nell’HTML”

# I Dogmi

## PHP

- ▶ **Ogni istruzione PHP termina con un punto e virgola**



# I Dogmi

PHP

- ▶ **Le condizioni o i parametri sono sempre tra parentesi tonde**
  - `if ($eta < 18)`
  - `while ($i > 0)`
  - `function somma( $num1, $num2 )`

# I Dogmi

## PHP

- ▶ **I blocchi di codice sono sempre tra parentesi graffe**
  - `if (...) { istruzione; istruzione; istruzione; }`
  - `while (...) { istruzione; istruzione; istruzione; }`
  - `function somma (...) { istruzione; istruzione; istruzione; }`
- ▶ **Eccezione**
  - Codice generale isolato, nell'HTML

# I Dogmi

PHP

- ▶ **Le funzioni ritornano qualcosa**
  - Terminano con `return VALORE;`
- ▶ **Le procedure non ritornano niente**
  - “Fanno solo cose”

# I Dogmi

PHP

- ▶ **Le chiamate a funzioni o procedure sono sempre nella forma**

**nomeFunzione(argomento1, argomento2, ...);**

**somma(1, 2);**

**somma(somma(1, 2), 3);**

# I Dogmi

PHP

- ▶ **I cicli che usano contatori necessitano di una istruzione di incremento (decremento)**

```
while ($i > 0) {
```

```
    ...
```

```
    $i--; // o $i++ a seconda della condizione
```

```
}
```

# I Dogmi

## Generale

- ▶ **I nomi delle cose in HTML, CSS, PHP hanno convenzioni**
  - Generalmente
    - tutto minuscolo
  - Unire più parole
    - HTML: sitienetuttominuscolo
    - CSS: si-separa-con-trattini
    - PHP: si\_usa\_trattino\_basso oppure SiUsaIlCamelCase
- ▶ **Rimanete consistenti**

# I Dogmi

## Generale

- ▶ **Siate curiosi**
- ▶ **Usate Google**
- ▶ **Iscrivetevi a forum e mailing list**

# I Dogmi

## Generale

- ▶ **Chiedetemi quando volete**