



Da zero a PHP...

Da zero a PHP...

- Se conosci un minimo di HTML (<div> <p> <a> <h1> <h2> <h3>) è un peccato non saper giocare in PHP (80% web, 100% WP)
- PHP = "**P**HP: **H**ypertext **P**reprocessor" (<https://www.php.net>); in poche parole, PHP genera pagine web dinamicamente (ovvero HTML!). E non solo.

Da zero a PHP...

- WP core, temi, plugin sono sviluppati in PHP. Se lo conosci... li customizzi. Bastano spesso 4 righe di codice per modificare/aggiungere qualcosa
- Prima di considerare l'installazione di *yet another plugin*, vedi se qualche riga di PHP può bastare :)

ZERO: le basi

Differenza tra HTML "manuale" e HTML generato da PHP

© 2011-2020 ACME INC.

© 2011-`<?php echo date('Y'); ?>` ACME INC.

Cosa fa PHP

- Genera HTML dinamico (date('Y') per esempio)
- Salva dati inseriti in un <form>
- Parla/scrive col/sul DB
- Parla/scrive con/su file
- Salva/legge cookies
- E molto altro

Sintassi

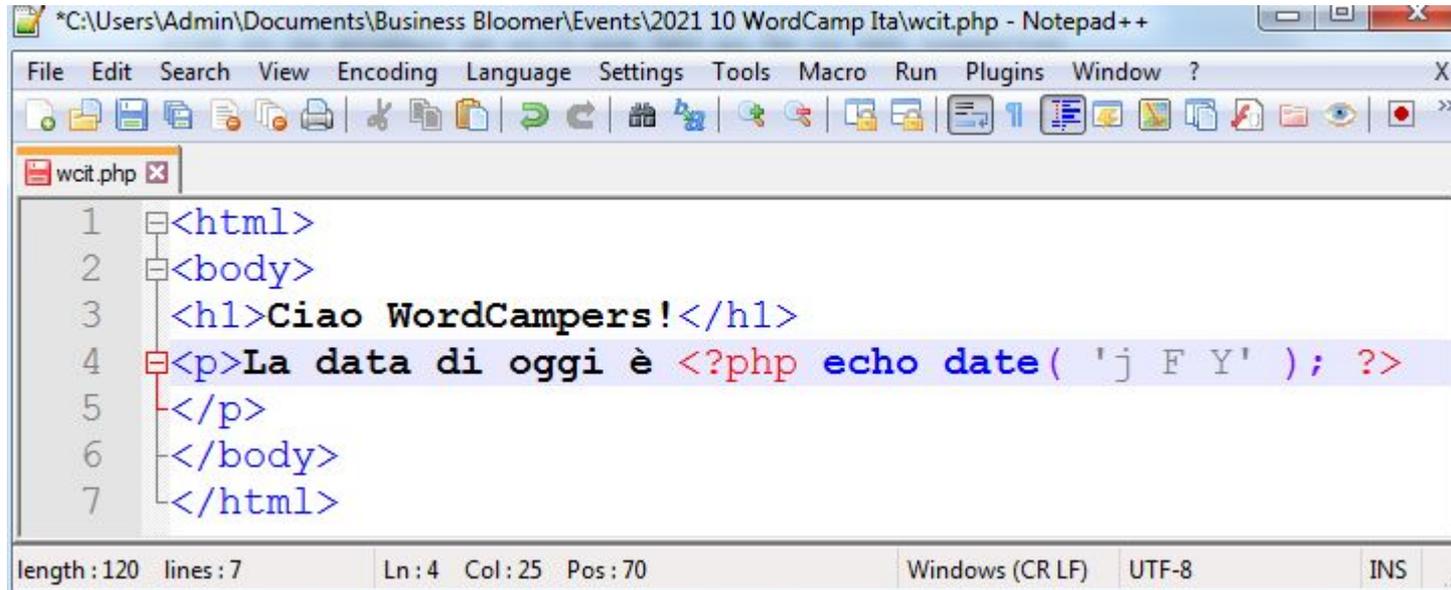
```
<?php echo date( 'Y' ); ?>
```

- PHP script è delimitato dai tag di apertura/chiusura
- PHP statement finiscono col ;
- PHP function date() ecc.
- PHP viene scritto in un file .php (functions.php per esempio)

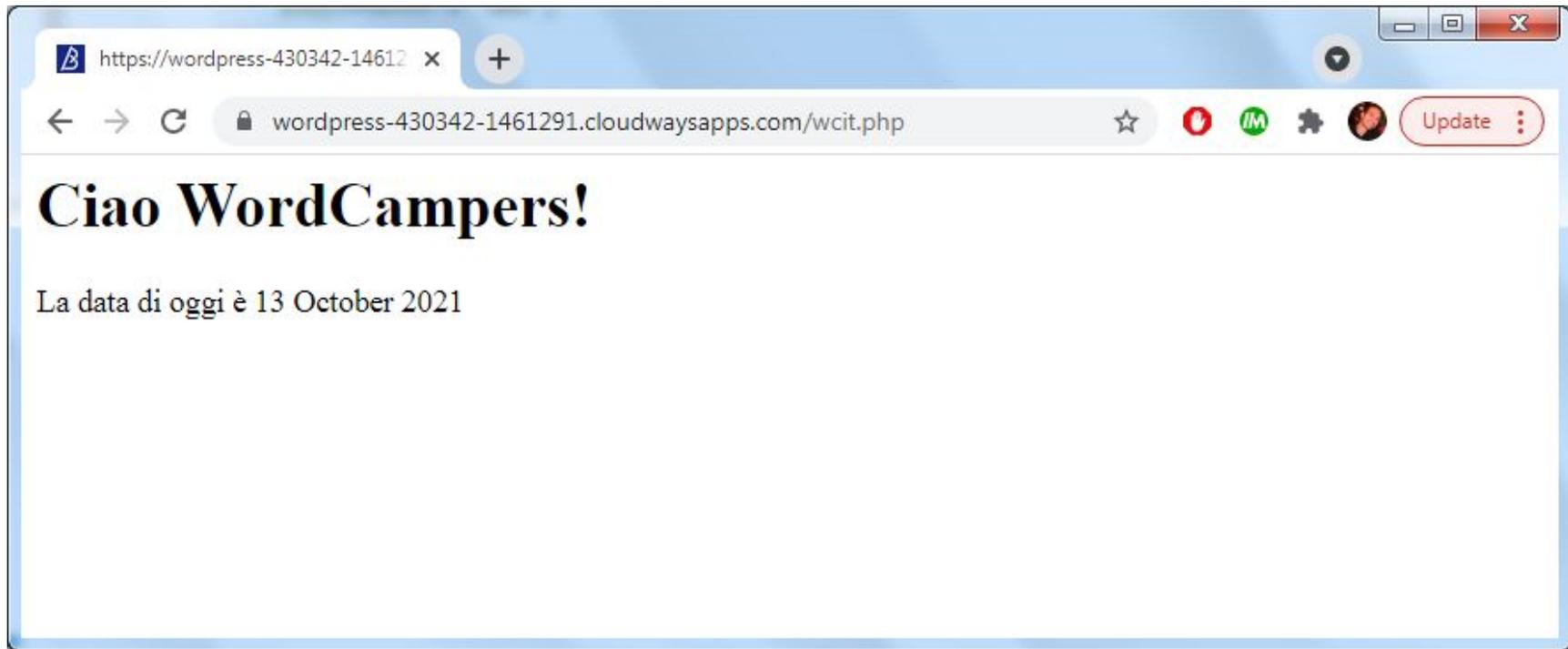
Prepararsi a scrivere in PHP

- Hosting con PHP installato (tutti praticamente!)
- Dominio
- SW: Code editor es. NotePad++
- SW: FTP es. FileZilla
- (sito WordPress)

Il nostro primo file .php (+ HTML)



```
*C:\Users\Admin\Documents\Business Bloomer\Events\2021 10 WordCamp Ita\wcit.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
wcit.php
1 <html>
2 <body>
3 <h1>Ciao WordCampers!</h1>
4 <p>La data di oggi è <?php echo date( 'j F Y' ); ?>
5 </p>
6 </body>
7 </html>
length : 120 lines : 7 Ln : 4 Col : 25 Pos : 70 Windows (CR LF) UTF-8 INS
```



UNO: *variabili*

Sintassi

```
<?php $totale = 59; ?>
```

In questo statement definisco la variabile (\$) \$totale e gli assegno il numero 59 e ne definisco il tipo (integer)

```
<?php $totale = "59"; ?>
```

?

Sintassi

```
<?php $totale = 0; ?>
```

- nome variabile inizia con lettera oppure "_"
- solo (A-z, 0-9, _)
- case-sensitive: \$Totale != \$totale

Tipi di variabile

```
<?php
```

```
$totale = 0; // integer
```

```
$totale = "n/a"; // string
```

```
$totale = 5.39; // float
```

```
$totale = array( 0, 1, 2, 7, 19 ); // array
```

```
$totale = false; // boolean
```

```
?>
```

Operatori

```
<?php
$totale = $x + $y;
$totale = $x * $y;
$totale = $x / $y;
$totale = $x . 'kg';
?>
```

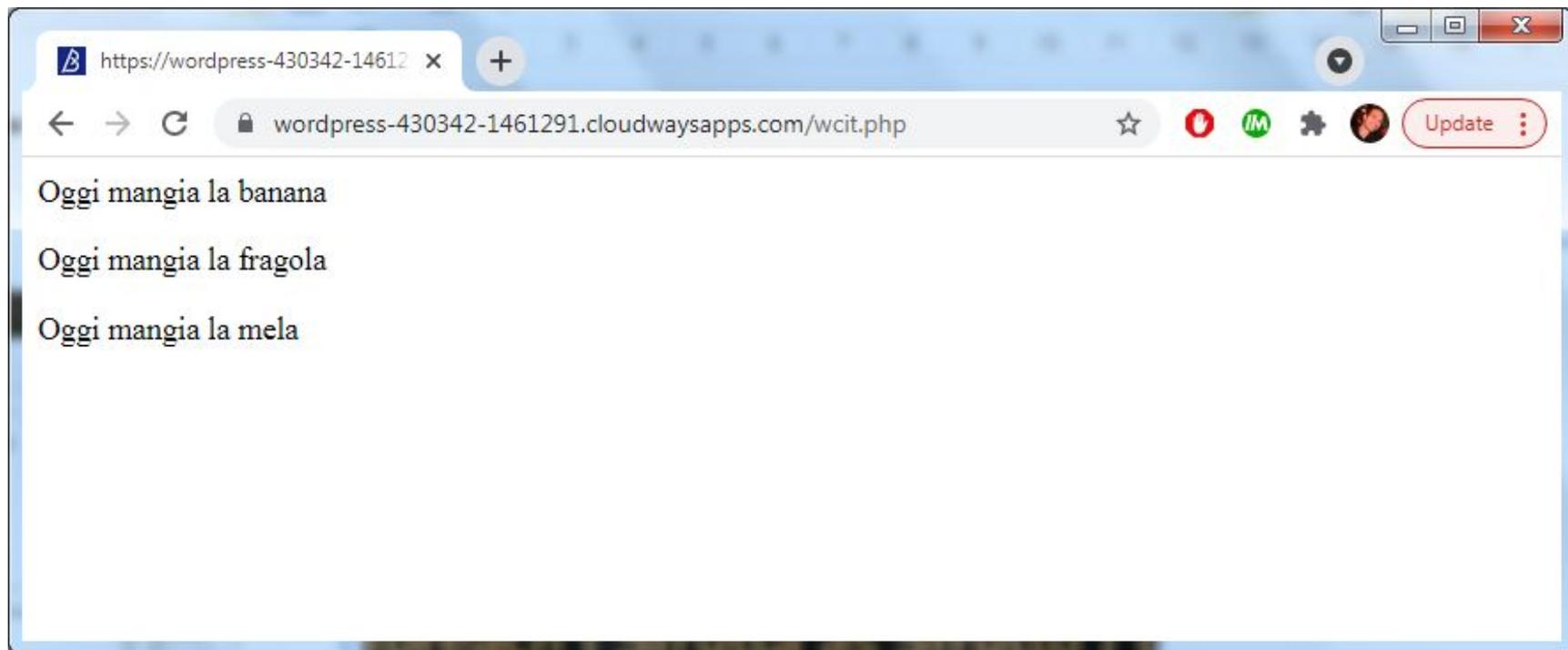
```
<?php
$x == $y // uguale
$x === $y // uguale + tipo
$x != $y // diverso
$x >= $y // maggiore uguale
?>
```

if else elseif Statement

```
<?php
if ( $totale == 0 ) {
    echo 'Nulla';
} elseif ( $totale >= 100 ) {
    echo 'Complimenti';
} else {
    echo 'Congratulazioni!';
}
?>
```

foreach Statement

```
<?php
$frutta = array( 'banana', 'fragola', 'mela' );
foreach ( $frutta as $frutto ) {
    echo '<p>Oggi mangia la ' . $frutto . '</p>';
}
?>
```



DUE: funzioni

PHP functions

- 1000+ built-in functions es. `date()` `strlen()`
- funzioni custom

Sintassi

```
<?php  
function hello() {  
    echo 'Ciao!';  
}  
hello(); // chiama  
?>
```

- nome funzione inizia con lettera oppure "_". Unico
- non è case-sensitive: HELLO();

Argomenti

```
<?php
function hello( $name ) {
    echo 'Ciao ' . $name . '!';
}
hello( 'Rodolfo' ); // chiama con argomento
?>
```

- si possono passare informazioni alla funzione
-

Argomenti

```
<?php  
function hello( $name = 'Pinco' ) {  
    echo 'Ciao ' . $name . '!';  
}  
hello( 'Rodolfo' ); // chiama con argomento  
?>
```

- si possono passare informazioni alla funzione
- **valore di default: in caso di assenza di argomento**

Echo vs Return

```
<?php
echo 'Ciao,'; // stampa Ciao,
function nome_cognome( $nome, $cognome ) {
    return $nome . ' ' . $cognome;
}
nome_cognome( "Rodolfo", "Melogli" ); // non stampa
?>
```

- echo = stampa; return = restituisci (no stampa)

Quindi a che serve Return?

```
<?php
function nome_cognome( $nome, $cognome ) {
    if ( $nome == "" || $cognome == "" ) return 'n/a';
    return $nome . ' ' . $cognome;
}
```

?>

- uscire presto dalla funzione

Quindi a che serve Return?

```
<?php
function nome_cognome( $nome, $cognome ) {
    if ( $nome == "" || $cognome == "" ) return 'n/a';
    return $nome . ' ' . $cognome;
}
if ( nome_cognome( "Rodolfo", "" ) == 'n/a' ) {
    echo 'Hey, manca qualcosa!';
}
?>
```

- uscire presto dalla funzione / **manipolare di nuovo**

TRE: PHP <> WP

PHP in WP

- Core, theme, plugin file sono quasi tutti .php
- Inoltre, WP definisce anche le sue funzioni PHP: *get_header()* *the_title()* *the_content()* ecc.
- Custom PHP si può inserire nel child theme per personalizzare WP/theme/plugin

WP Hooks

- Sito WP è un assemblaggio di funzioni PHP che poi restituiscono il risultato nel browser (header, contenuto, sidebar, footer...)
- Durante questo assemblaggio, puoi "stoppare" il processo, "agganciare" la tua funzione, e farlo ripartire SENZA modificare i file WP/theme/plugin

WP Hooks - Esempio "wp_head"

```
<?php
add_action( 'wp_head', 'googleanalytics' );
function googleanalytics() {
    echo '<script>codice qui</script>';
}
?>
```

- mentre WP assemblava l'header, ho "agganciato" il mio script chiamando la mia funzione custom

WP Hooks - Esempio "the_content"

```
<?php
add_filter( 'the_content', 'concatena_banner' );
function concatena_banner( $content ) {
    $content = $content . '';
    return $content;
}
?>
```

- mentre WP assemblava il contenuto, ho "agganciato" il banner chiamando la mia funzione custom

WP Hooks: `add_action` `add_filter`

- *add_action* ti permette di agganciarti a una *do_action* per aggiungere qualcosa (contenuto, un'azione, ecc.)
- *add_filter* ti permette di agganciarti a un *apply_filters* per manipolare qualcosa (es. array di categorie)

Un plugin ben sviluppato ti offre:

- una marea di *do_action* per aggiungere personalizzazioni tramite `functions.php`
- una marea di *apply_filters* per intercettare le sue funzioni e manipolarle tramite `functions.php`
- altrimenti devi modificare i core files (big NO)

Caso Studio 1: prefooter

- su tutte le pagine WP si desidera aggiungere una sezione "prefooter" con un banner
- vado a cercarmi l'hook di WP che corrisponde al momento in cui assembla il footer
- mi "aggancio" e faccio caricare prima la mia funzione custom

Caso Studio 1: prefooter

```
<?php
add_action( 'get_footer', 'prefooter', 1 );
function prefooter() {
    echo '';
}
?>
```

priority





Animals

Caso Studio 2: rimuovere WC tabs

- WooCommerce "aggiunge" i single product tabs con un suo hook: `add_action('woocommerce_after_single_product_summary', 'woocommerce_output_product_data_tabs', 10);`
- tramite PHP vogliamo rimuoverlo
- businessbloomer.com/woocommerce-visual-hook-guide-single-product-page per lista hooks

Caso Studio 2: rimuovere WC tabs

unica differenza

```
<?php  
remove_action(  
'woocommerce_after_single_product_summary',  
'woocommerce_output_product_data_tabs', 10 );  
?>
```



Caso Studio 3: SEO

```
<?php
function home_meta_description() {
    if ( is_home() ) {
        echo '<meta name="description" content="Il
WordCamp Italia è un evento online organizzato dalla
community italiana di WordPress">';
    }
}
add_action( 'wp_head', 'home_meta_description');
?>
```

Da zero a PHP

Recap

- Potremmo fare un WordCamp intero a riguardo...
- In questi minuti hai imparato a scrivere in PHP, creare funzioni e manipolare un sito WP
- Ora sta a te continuare con lo studio e la pratica. Se in 15 minuti abbiamo fatto tutto questo, chissà in 100 ore cosa si può ottenere!



Grazie!
Domande?