

Lavorare con i file

Corso Backend System Integrator
Modulo **Programmazione PHP**

Docente: Dott. Enrico Zimuel

in collaborazione con:



REGIONE
PIEMONTE

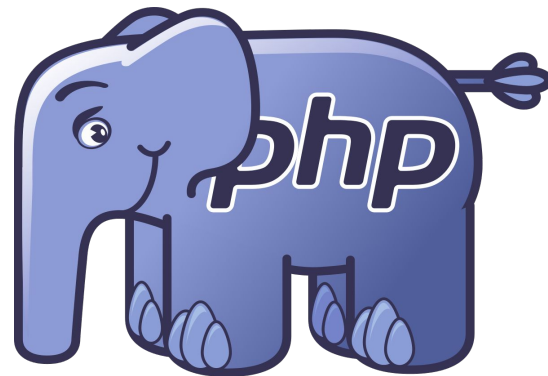
per una crescita intelligente,
sostenibile ed inclusiva

www.regione.piemonte.it/europa2020

INIZIATIVA CO-FINANZIATA CON FSE

Programma

- Leggere il contenuto di un file
- Scrivere in un file
- Verificare se un file esiste
- Eliminare un file
- Leggere un file in formato CSV



Leggere il contenuto di un file

- In PHP ci sono molti modi per leggere il contenuto di un file:
 - [file_get_contents\(\)](#)
 - [file\(\)](#)
 - [fopen\(\)](#) + [fgets\(\)](#) + [fclose\(\)](#)
- In generale il PHP mette a disposizione [molte funzioni](#) per la gestione dei file

file_get_contents()

- La funzione [file_get_contents\(\)](#) legge il contenuto di un file e lo memorizza in una stringa

```
// esempio.txt  
Hello World!
```

```
$contents = file_get_contents ('esempio.txt');  
echo $contents; // Hello World!
```

- Legge tutto il contenuto del file, quindi se il file è grande ci può essere un problema di memoria

file()

- La funzione [file\(\)](#) legge il contenuto di un file e lo memorizza ogni riga in un array

```
// esempio.txt
```

```
Linea 1
```

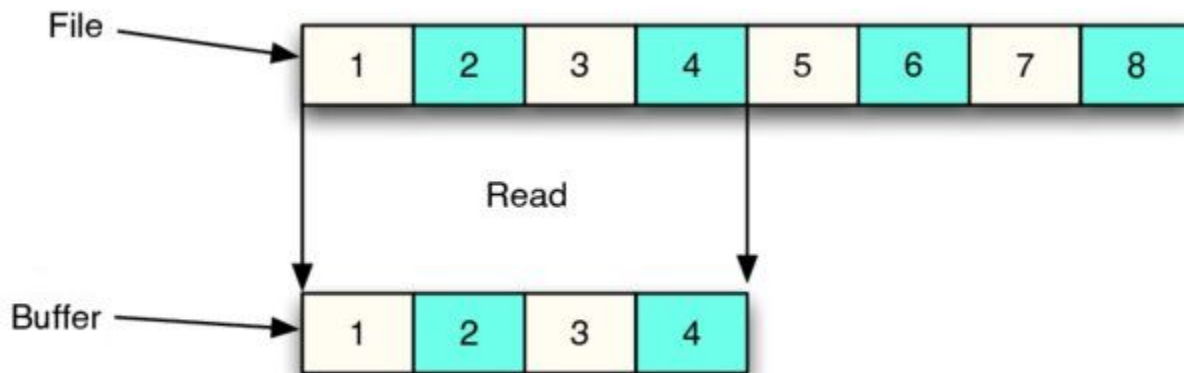
```
Linea 2
```

```
Linea 3
```

```
$contents = file('esempio.txt');  
var_dump($contents);  
/*  
array(3) {  
    [0]=> string(8) "Linea 1"  
    [1]=> string(8) "Linea 2"  
    [2]=> string(7) "Linea 3"  
}  
*/
```

fopen(), fgets(), fclose()

- E' possibile leggere il contenuto di un file un "pezzo alla volta" tramite l'utilizzo di un **buffer di memoria**



fgets()

- Con [fgets\(\)](#) è possibile leggere una riga alla volta (fino a EOL)

```
$file = fopen("esempio.txt", "r");  
while(($data = fgets($file)) !== false) {  
    var_dump($data);  
}  
fclose($file);
```

```
string(8) "Linea 1"  
"  
string(8) "Linea 2"  
"  
string(7) "Linea 3"
```

fgets() con dimensione del buffer

- E' possibile specificare la dimensione del buffer in fgets(\$file, int \$length)
- fgets() leggerà fino a **\$length-1** caratteri

```
$file = fopen("esempio.txt", "r");  
while(($data = fgets($file, 5)) !== false) {  
    var_dump($data);  
}  
fclose($file);
```

```
string(4) "Line"  
string(4) "a 1"  
"  
string(4) "Line"  
string(4) "a 2"  
"  
string(4) "Line"  
string(3) "a 3"
```


Scrivere in un file

- E' possibile scrivere in un file utilizzando la funzione [file_put_contents\(\)](#)

```
file_put_contents('test.txt', 'Hello World!');
```

- Si può utilizzare anche la funzione [fwrite\(\)](#), avendo una risorsa file aperta con fopen()

fwrite()

```
$file = fopen("dati.csv", "w");  
for ($riga=0; $riga<10; $riga++) {  
    for($i=0; $i<10; $i++) {  
        fwrite($file, rand(0,100) . ';');  
    }  
    fwrite($file, PHP_EOL);  
}  
fclose($file);  
echo file_get_contents("dati.csv");
```

```
// dati.csv  
9;35;96;16;2;78;85;67;26;63;  
93;13;48;39;45;53;22;95;20;85;  
32;79;72;18;57;85;13;39;79;28;  
99;59;2;82;66;9;79;59;67;42;  
69;30;59;72;53;69;19;67;38;45;  
12;73;52;11;70;82;69;3;72;0;  
83;45;91;38;84;89;84;60;11;64;  
64;99;75;58;75;87;53;79;55;26;  
93;4;85;21;81;19;67;85;62;4;  
48;2;49;31;82;71;71;11;17;54;
```

Verificare se un file esiste

- In PHP è possibile verificare se un file esiste con la funzione [file_exists\(\)](#)

```
if (file_exists("dati.csv")) {  
    echo "Il file CSV esiste";  
} else {  
    echo "Il file CSV non esiste :-( ";  
}
```

Eliminare un file

- Per eliminare un file è necessario utilizzare la funzione un [unlink\(\)](#)

```
unlink("dati.csv");  
  
if (!file_exists("dati.csv")) {  
    echo "Il file CSV non esiste :-(" ;  
}
```

Leggere un file CSV

- E' possibile leggere un file csv in diversi modi, ad esempio utilizzando la funzione [fgetcsv\(\)](#)

```
$file = fopen("dati.csv", "r");  
while(($data = fgetcsv($file, 100, ";")) !== false) {  
    var_dump($data);  
}  
fclose($file);
```

```
// dati.csv  
9;35;96;16;2;78;85;67;26;63;  
...
```

```
[9, 35, 96, 16, 2, 78, 85, 67, 26, 63, PHP_EOL]  
...
```

Esercizio (da consegnare)

- Leggere [questo file CSV](#) e creare un altro file calcolando la media aritmetica di ogni riga
- L'output dovrà essere un file dove per ogni riga è riportata la media della corrispondente riga del file CSV
- Le media aritmetica deve essere riportata con 4 cifre decimali

Esercizio di preparazione per verifica

- Da una serie di sensori IoT sono stati raccolti dei dati dal 14 al 18 maggio 2023
- I dati sono dei numeri interi compresi tra 0 e 100
- I dati sono stati raccolti in dei [file CSV](#) suddivisi per ogni giorno
- Creare una classe denominata **Foglio** per analizzare un file CSV:
 - leggere il file CSV
 - calcolare la media di ogni riga del file
 - calcolare la media di tutti i dati presenti nel file
- Creare una classe denominata **Collezione** per gestire gli oggetti **Foglio**:
 - calcolare la media di tutti i fogli

Grazie dell'attenzione!

Per informazioni:

enrico.zimuel@its-ictpiemonte.it