

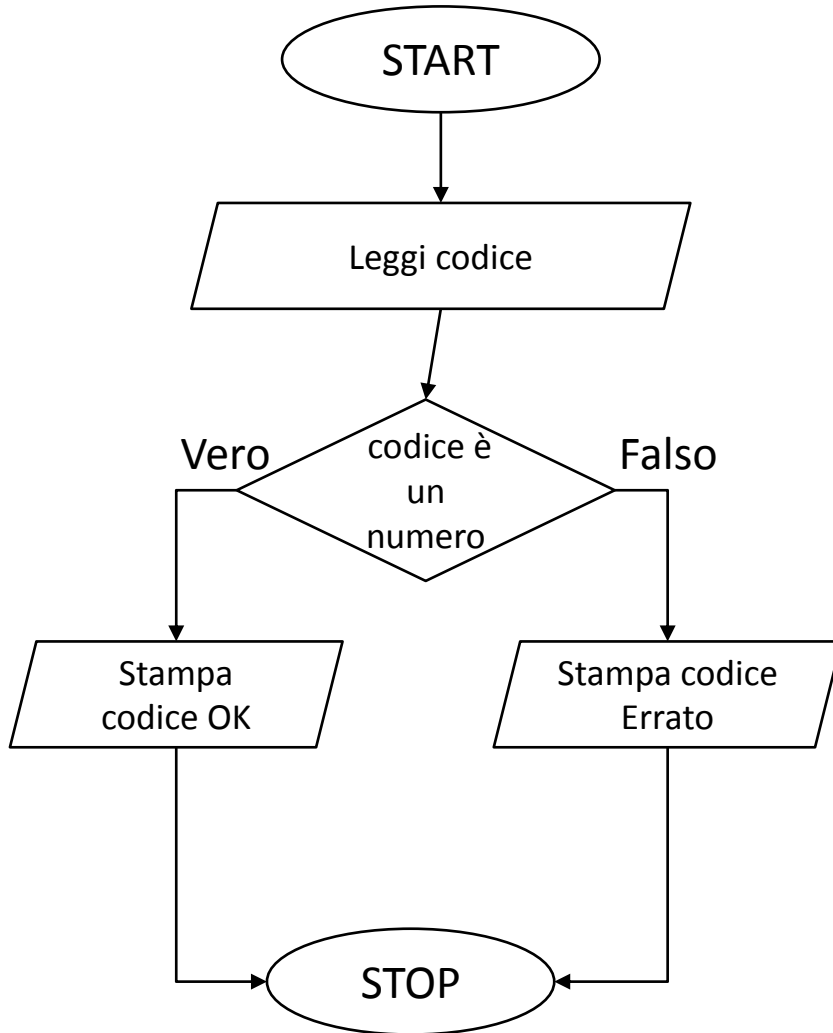
# Cicli WHILE in PHP

# Problema

Scrivere un programma PHP che “prelevato” un codice da una FORM verifica se tale codice è un numero.

In particolare, stampare un messaggio di errore se la stringa contenuta nella variabile *codice* non è un numero.

# Controllo Codice



```
<?php
$cod = $_POST['codice'];
$codIsNum = is_numeric($cod);
if ($codIsNum == true)
{
    echo "Codice OK";
}
else
{
    echo "codice ERRATO";
}
?>
```

# Controllo Codice

come viene eseguito dal computer

INPUT="125"

```
<?php
$cod = 125
$codIsNum = true → (RISULTATO di is_numeric(125))
if (true == true) → VERO
{
    echo "Codice OK";
}
else
{
    echo "codice ERRATO";
}
?>
```

→ NON VIENE ESEGUITO

**OUTPUT stampa codice OK**

# Controllo Codice

come viene eseguito dal computer

INPUT="a125"

```
<?php
$cod = a125
$codIsNum = false → (RISULTATO di is_numeric(a125))
if (false == true) → FALSO
{
    echo "Codice OK";
}
else
{
    echo "codice ERRATO";
}
?>
```

NON VIENE ESEGUITO

**OUTPUT stampa codice ERRATO**

# Problema

Scrivere un programma PHP che stampi ogni carattere di un  
**codice di esattamente 3 caratteri**  
su una nuova riga.

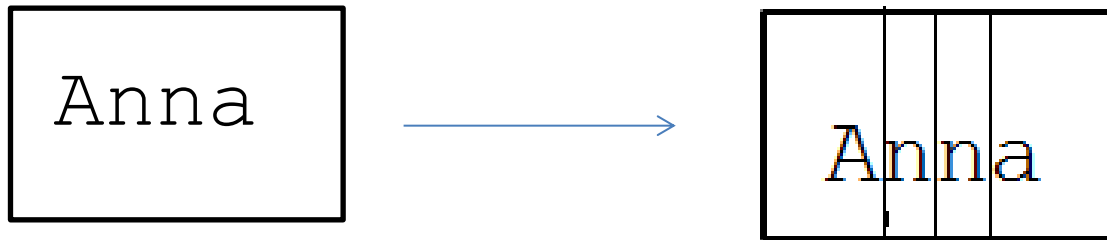
ESEMPIO: Se codice = 123 allora stampare:

1

2

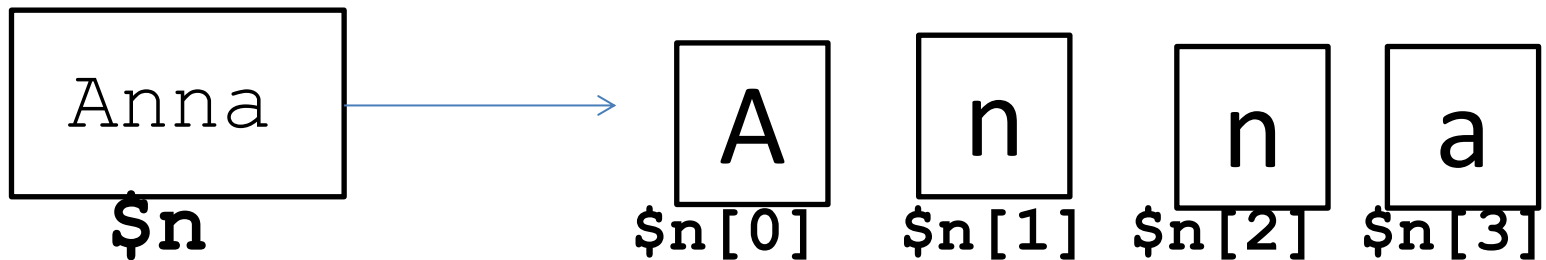
3

Ogni variabile contenente una stringa può essere suddivisa in 'sottovariabili' che contengono esattamente N°1 caratteri



**\$n**

Ogni variabile contenente una stringa può essere suddivisa in 'sottovariabili' che contengono esattamente N°1 caratteri



Ogni sottovariabile viene identificata con il nome della variabile seguito dalla numero della posizione tra parentesi quadre.



**variabile[numero]** restituisce il singolo carattere di una stringa corrispondente alla posizione **numero**

Le cifre si contano a partire da 0

# Stampa codice di tre cifre su più righe

Scrivere un programma PHP che stampi ogni carattere di un codice di  
**esattamente 3 carattere**  
su una nuova riga.

ESEMPIO: Se codice = 123 allora stampare:

1

2

3

# Stampa codice di tre cifre su più righe

```
<?php
$cod = trim($_POST['codice']);
echo "$cod[0] <BR>";
echo "$cod[1] <BR>";
echo "$cod[2] <BR>";
?>
```

# Stampa l'ultima cifra di un codice di lunghezza generica

Scrivere un programma PHP che stampi l'ultimo carattere di un codice di **lunghezza generica**

ESEMPIO:

Se codice = 123 allora stampare: **3**

Se codice = 123498 allora stampare: **8**

Se codice = 1 allora stampare: **1**

# Stampa codice di tre cifre su più righe

```
<?php
$cod = trim($_POST['codice']);
$lunghezza = strlen($cod);
$ultimo_carattere = $lunghezza - 1;
echo "$cod[$ultimo_carattere]";
?>
```

# Parentesi quadre variabile

**variabile[numero]** restituisce il singolo carattere di una stringa corrispondente alla posizione **numero**

- Le cifre si contano a partire da 0
- L'ultima posizione è data dalla «lunghezza – 1»

**ATTENZIONE:** per lunghezza intendiamo quella ripulita dagli spazi superflui

# Problema

Scrivere un programma PHP che stampi ogni carattere di un  
**codice di lunghezza generica**  
su una nuova riga.

ESEMPIO: Se codice = 1ab45 allora stampare:

1

a

b

4

5

# While

**SINTASSI:** `while (condizione) {istruzione1; istruzione2;  
... }`

I cicli `while` sono la forma più generica di ciclo. La struttura generale è questa:

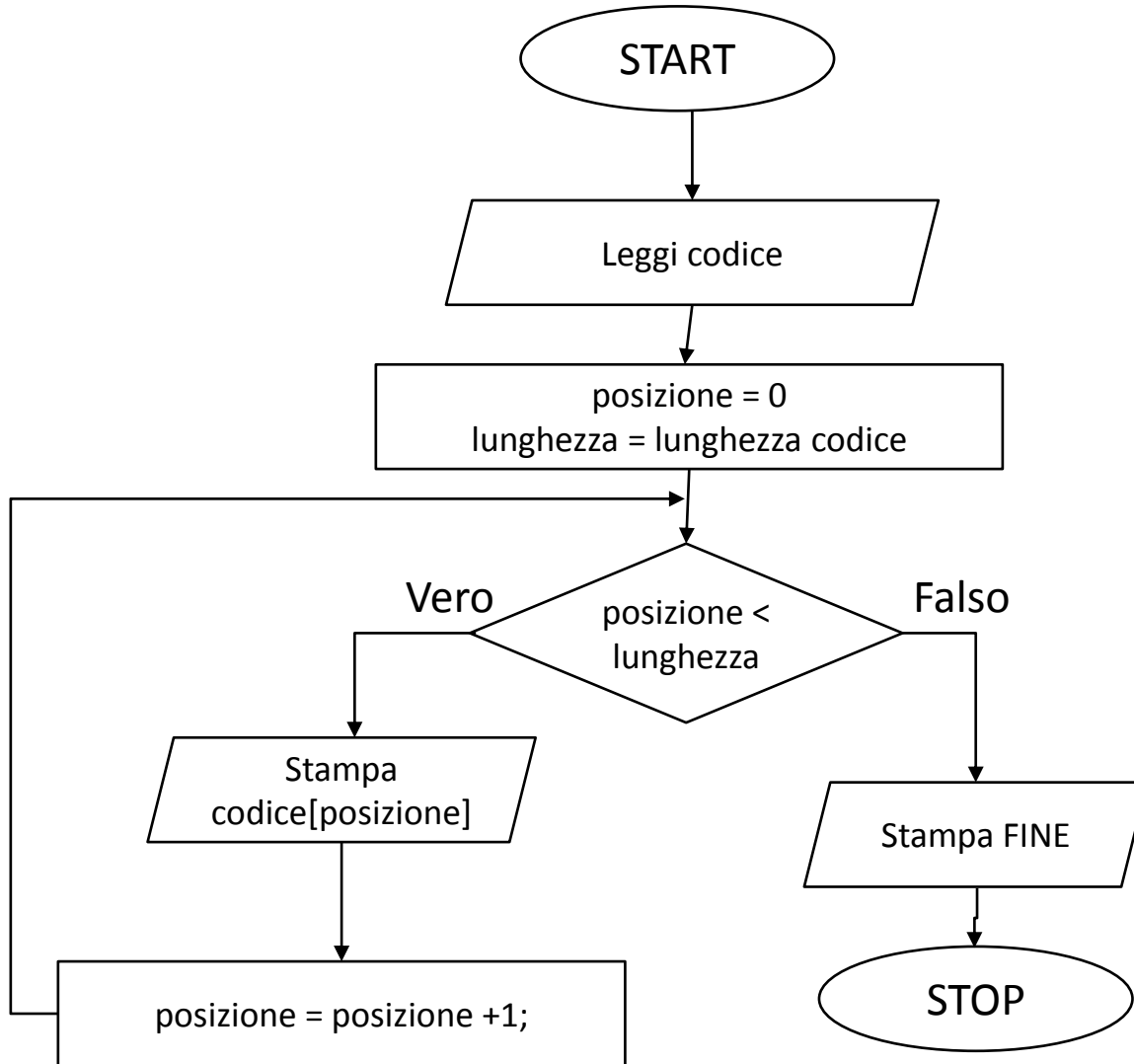
Quando viene eseguito, si compiono i seguenti passi:

1. si verifica se la condizione è vera;
2. se lo è, si eseguono in sequenza le istruzioni `istruzione1`, `istruzione2`, ecc. e si verifica di nuovo la condizione

Il ciclo si ripete fino a quando `(condizione)` del *while* non diventa falsa



# Stampa i caratteri di Codice



# Stampa i caratteri di Codice

```
<?php
$cod = trim($_POST['codice']);
$posizione = 0;
$lunghCod = strlen($cod);
while ($posizione < $lunghCod)
{
    echo "$cod[$posizione]<BR>";
    $posizione = $posizione + 1;
}
echo "FINE";
?>
```

# Stampa i caratteri di Codice

come viene eseguito dal computer

INPUT="a125"

```
<?php
$cod = a125
$posizione = 0;
$lunghezzaCod = 4 → RIS strlen($cod)
while (0 < 4) → VERA
{
    echo "a <BR>";
    $posizione = 1 → RIS 0 + 1;
}
TORNA AL WHILE
while (1 < 4) VERA
{
    echo " 1 <BR>";
    $posizione = 2 → RIS 1 + 1;
}
TORNA AL WHILE
```

```
while (2 < 4) VERA
{
    echo "2<BR>";
    $posizione = 3 → RIS 2+1
}
TORNA AL WHILE
while (3 < 4) VERA
{
    echo "5<BR>";
    $posizione = 4 → RIS 3+1;
}
TORNA AL WHILE
while (4 < 4) FALSO
{} NON E' ESEGUITO
echo "FINE";
```