

# Esercizi - 003

**TIPS 1:** Per migliorare l'approccio alle funzioni, il consiglio è sempre di risolvere la traccia utilizzando come prima cosa l'approccio sequenziale. Successivamente procedere con:

1. Estrazione
2. Incapsulamento
3. Astrazione

**TIPS 2:** Controllare sulla documentazione ufficiale di PHP.NET se esiste una Built-In function da utilizzare per il caso specifico.

---

## ESERCIZIO 1

Utilizzando un approccio funzionale (piccole funzioni che risolvono un piccolo problema), riscrivere il programma fatto in precedenza che stampi in console tutti i numeri da 1 a 30.

- Se il numero è multiplo di 3, non deve stampare il numero ma "PHP";
  - Se multiplo di 5 deve stampare "JAVASCRIPT";
  - Se multiplo di 3 e 5 (15) deve stampare "Bari";
- 

## ESERCIZIO 2

Dato il seguente array associativo:

```
$persons = [  
  [  
    'name' ⇒ 'Marco',  
    'age' ⇒ 18,  
  ],  
  [  
    'name' ⇒ 'Vanessa',  
    'age' ⇒ 12,  
  ],  
]
```

```
],  
[  
  'name' ⇒ 'Jack',  
  'age' ⇒ 34,  
],  
[  
  'name' ⇒ 'Maria',  
  'age' ⇒ 55,  
],  
];
```

Scrivere un programma utilizzando le funzioni in grado di controllare l'età di ogni persona e stampare a video se:

- Maggiore o uguale di 18: "\$nome\_persona può votare"
- Minore di 18: "\$nome\_persona non può votare"

## ESERCIZIO 3

Scrivere un programma funzionale che, dato un numero in input (\$max), stampi a video:

- PRIMA tutti i numeri Dispari
- DOPO tutti i numeri PARI

```
//Esempio max = 10  
1,3,5,7,9  
0,2,4,6,8,10
```

## Esercizio 4

Scrivere un programma che, data una stringa in input dal readline, conti quante vocali sono state inserite

```
//Esempio  
  
Aiuole = Nella parola "Aiuole" ci sono 5 vocali
```

Alba = Nella parola "Alba" ci sono 2 vocali  
Fgrty = Nella parola "Fgrty" ci sono 0 vocali

## Esercizio 5

Dato il seguente array associativo:

```
$persons = [  
  [  
    'name' ⇒ 'Marco',  
    'age' ⇒ 20,  
    'gender' ⇒ "M"  
  ],  
  [  
    'name' ⇒ 'Vanessa',  
    'age' ⇒ 10,  
    'gender' ⇒ "F"  
  ],  
  [  
    'name' ⇒ 'Jack',  
    'age' ⇒ 35,  
    'gender' ⇒ "M"  
  ],  
  [  
    'name' ⇒ 'Maria',  
    'age' ⇒ 55,  
    'gender' ⇒ "F"  
  ],  
  [  
    'name' ⇒ 'Enrica',  
    'age' ⇒ 35,  
    'gender' ⇒ "F"  
  ],  
];
```

Calcolare:

- L'età media di tutte le persone con genere M

- L'età media di tutte le persone con genere F

Da specifiche, saranno solamente 2 (booleani) i possibili valori di gender: M e F.

Creare una funzione che sia in grado di operare nel modo più astratto possibile:

- Verificare il sesso
- Aggiungere +1 a Maschio o femmina
- E sommi di volta in volta l'età