

PROGETTO JAVASCRIPT BASE

A L E S S A N D R O A G L I A N O '



IN COSA CONSISTE?

Il progetto consiste nel creare un contatore che tramite due pulsanti incrementi o diminuisca il suo valore.

Ho voluto aggiungere i pulsanti **Reset** e **Mode** per implementare più funzionalità

Counter

0

+

Reset

-

Mode

Produced by [Alessandro Aglianò](#)





CREAZIONE ELEMENTI

Alla funzione 'createElement' ho inserito dei parametri: tag dell'elemento, classe, il genitore dove appendere il div e un eventuale testo.

infine ho ritornato il div come valore

```
function createElement(tagname, classe, genitore, text) {  
  tagname = document.createElement(tagname);  
  tagname.classList.add(classe);  
  genitore.appendChild(tagname);  
  text = text || ""; // il valore vuoto è per evitare la scritta 'undefined'  
  var text = (tagname.innerHTML = text);  
  return tagname;  
}  
  
const counter = createElement("div", "counter", mainContainer); // DIV COUNTER  
const title = createElement("div", "title", counter, "Counter"); // Titolo Counter  
const display = createElement("div", "display", counter, "0"); // Display del contatore  
const buttonContainer = createElement("div", "button-container", counter); //CONTAINER PER I BOTTONI  
const addButton = createElement("div", "button", buttonContainer, "+"); // Bottone Aggiungi  
const resetButton = createElement("div", "button", buttonContainer, "Reset"); // Bottone Reset  
const decreaseButton = createElement("div", "button", buttonContainer, "-"); // Bottone Sottrai  
const modeButton = createElement("div", "button", buttonContainer, "Mode"); // Bottone Mode
```

I Pulsanti



+

Aggiungi

-

Sottrai

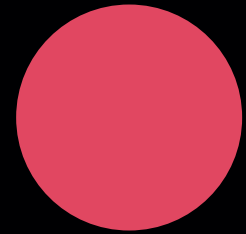
reset

Reset

mode

Mode





TASTO AGGIUNGI

per il pulsante **Aggiungi** ho creato 2 funzioni :

- **addCounterValue()** = serve ad incrementare il valore del contatore ad ogni click del pulsante .
- **loopAdd()** = serve ad incrementare il valore del contatore in modo esponenziale quando si tiene premuto il pulsante.

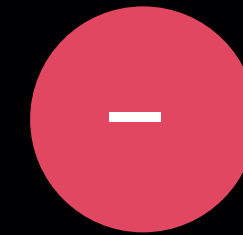
Questo loop l'ho creato grazie a **SetTimeout** e a **SetInterval**.

```
let valueOfCounter = 0;
let time;
let loop;

const loopAdd = function loopToAddCounterValue() {
  addCounterValue();
  time = setTimeout(() => {
    loop = setInterval(addCounterValue, 30);
  }, 1500);
};

const addCounterValue = function () {
  valueOfCounter += 1;
  display.innerHTML = valueOfCounter;
};

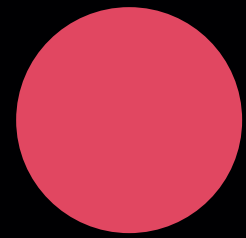
const stopLoop = function () {
  clearTimeout(time);
  clearInterval(loop);
};
```

TASTO SOTTRAI

Il tasto **sottrai** segue la stessa logica del tasto Aggiungi con la differenza che per la funzione 'decreaseCounterValue()' ho impostato la condizione che il *contatore sia minore di zero* ($\text{contatore} < 0$) per fare in modo che il valore non sia mai negativo.

```
const loopDecrease = function loopToDecreaseCounterValue() {  
  decreaseCounterValue();  
  time = setTimeout(() => {  
    loop = setInterval(decreaseCounterValue, 30);  
  }, 1500);  
};  
  
const decreaseCounterValue = function () {  
  if (valueOfCounter > 0) {  
    valueOfCounter -= 1;  
  }  
  display.innerHTML = valueOfCounter;  
};
```



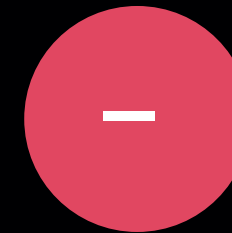
TASTO RESET

Per il pulsante **Reset** ho creato la funzione 'resetCounter()' che riporta il valore del contatore= 0 e stampa questo valore sullo schermo.

```
const resetCounter = function () {  
  valueOfCounter = 0;  
  display.innerHTML = valueOfCounter;  
};
```



```
function changeColor() {  
  if (body.className == "primary-color") {  
    body.classList.remove("primary-color");  
    body.classList.add("secondary-color");  
  } else {  
    body.classList.remove("secondary-color");  
    body.classList.add("primary-color");  
  }  
}
```



TASTO MODE

Il tasto **Mode** serve a cambiare l'interfaccia grafica tra bianco e nero.

E' stata creata alternando al click la classe *'primary-color'* e *'secondary-color'*.


```

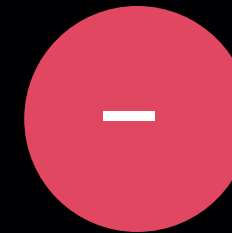
const clickButton = function (event) {
  if (event.target == resetButton) {
    resetCounter();
  } else if (event.target == modeButton) {
    changeColor();
  }
};

const pointerDownButton = function (event) {
  if (event.target == addButton) {
    loopAdd();
  } else if (event.target == decreaseButton) {
    loopDecrease();
  }
};

const pointerUpButton = function (event) {
  if (event.target == addButton || event.target == decreaseButton) {
    stopLoop();
  }
};

const pointerOutButton = function (event) {
  if (event.target == addButton || event.target == decreaseButton) {
    stopLoop();
  }
};

```



FUNZIONI PRINCIPALI

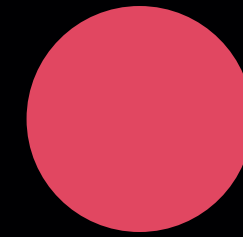
Infine ho creato 4 funzioni per ogni event.type : click, pointerdown, pointerUp, pointerOut che collego al dip wrapper dei bottoni.

In ogni funzione c'è la condizione per quale event.target si attiva.

```

buttonContainer.addEventListener("click", clickButton);
buttonContainer.addEventListener("pointerdown", pointerDownButton);
buttonContainer.addEventListener("pointerup", pointerUpButton);
buttonContainer.addEventListener("pointerout", pointerOutButton);

```



LINK UTILI

- **Counter:**
<https://counter-agliano.netlify.app/>
- **Sito personale:**
<https://alessandroagliano.github.io/>
- **Repository progetto github:**
<https://github.com/alessandroagliano/Counter-App>

