JavaScript Asincrono

Le Callback function

Paolo Caramanica

Sommario

- Le funzioni
  - Le funzioni
  - Definizione delle funzioni in JavaScript
  - Caratteristiche delle funzioni in JavaScript
  - Live Demo
- Le operazioni asincrone
  - Le funzioni asincrone
  - Primi esempi di funzioni asincrone
  - Live Demo
- Single thread
  - Single thread ed operazioni asincrone
  - JavaScript nel browser
  - Live Demo
  - JavaScript nel browser – esempio
  - Una precisazione su setTimeout

Le funzioni

Particolarità delle funzioni in JavaScript

Le funzioni

- Sono blocchi di codice, identificati da un nome, che possono essere richiamati in altri punti del programma
  - Possono accettare dei parametri in input
  - Possono produrre un output (valore di ritorno)
  - Permettono il riuso del codice
- Esempio:

```
function AreaQuadrato(lato) {
    return lato*lato;
}
```

Definizione delle funzioni in JavaScript

- Function definition:

```
function AreaQuadrato(lato) {
    return lato*lato;
}
```

- Function expression:

```
let AreaQuadrato = function(lato) {
    return lato*lato;
}
```

- Arrow function:

```
let AreaQuadrato = (lato) => lato*lato;
```

Caratteristiche delle funzioni in JavaScript

- Le funzioni in JavaScript:
  - Possono essere assegnate ad una variabile
  - Possono essere definite dentro un'altra funzione
  - Possono essere passate come argomento ad un'altra funzione (**callback**)
  - Possono essere restituite, come valore di ritorno, da un'altra funzione

Live demo

Le operazioni asincrone

Implementazione di funzioni asincrone con le callback

Le funzioni asincrone

- Se un'elaborazione richiede un tempo di attesa per qualche evento esterno, ad esempio la risposta da un server:
  - Non è efficiente bloccare lo script
  - Conviene implementarla in modo asincrono
- Le funzioni asincrone:
  - Non blocano lo script principale
  - Non restituiscono valori con return
  - Terminata l'elaborazione, chiamano una funzione di callback

Primi esempi di funzioni asincrone

- setTimeout:
  - E' una funzione built-in di JavaScript
  - Accetta due parametri:
    - Una funzione callback, che verrà eseguita una sola volta, dopo un dato intervallo di tempo
    - L'intervallo di tempo, espresso in millisecondi, dopo il quale eseguire la callback
- setInterval:
  - E' una funzione built-in di JavaScript
  - E' Simile a SetTimeout, ma esegue la funzione di callback ad ogni intervallo

Live demo

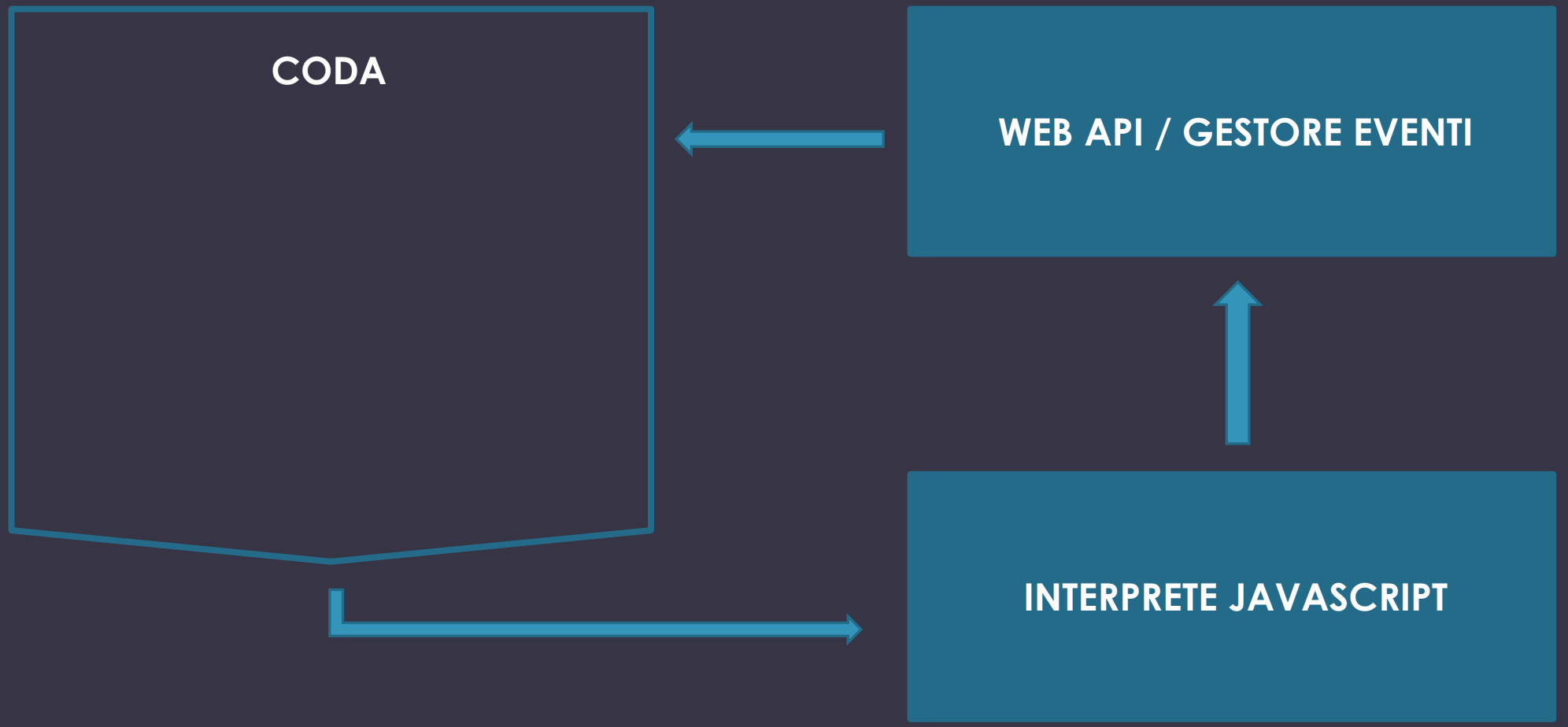
Single Thread

Single thread e programmazione asincrona

Single Thread ed operazioni asincrone

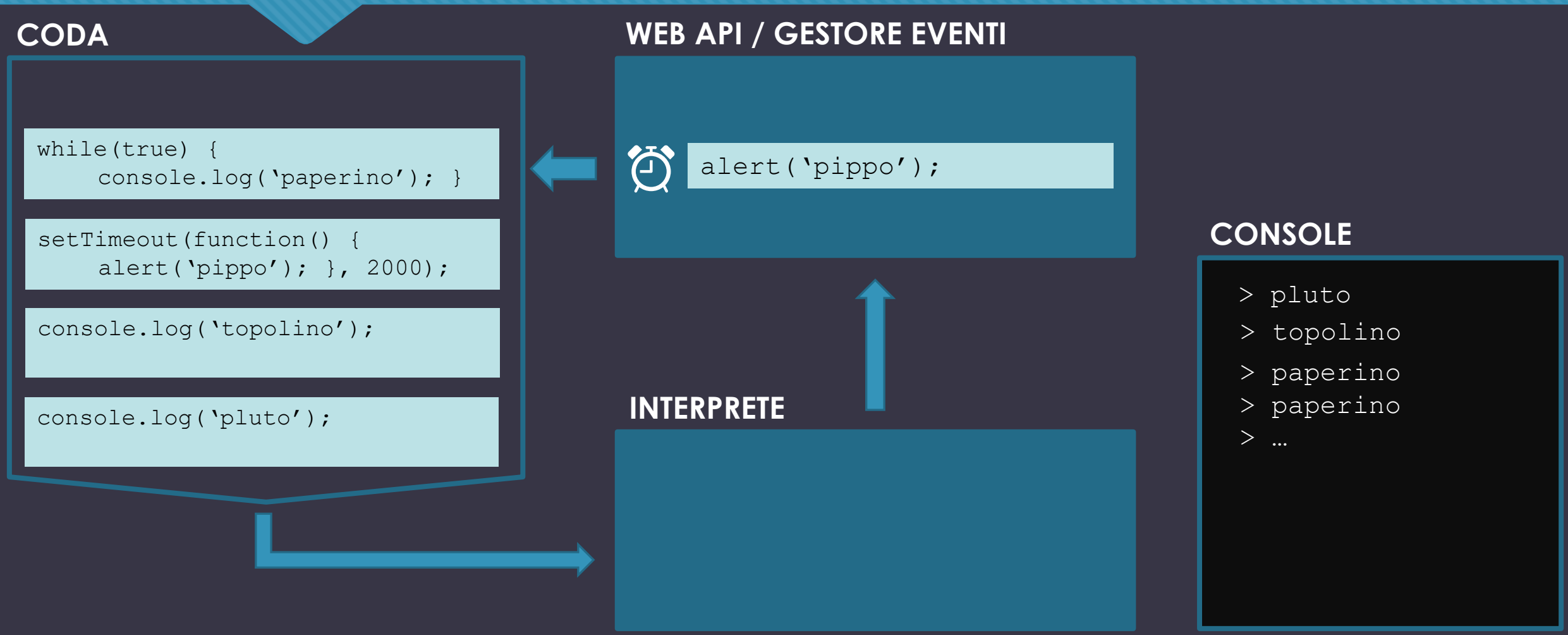
- JavaScript è un linguaggio single thread:
  - Le istruzioni da eseguire sono inserite in un'unica coda
  - C'è un solo interprete, che ne esegue una alla volta
  - Solo dopo aver completato un'istruzione, può passare a quella successiva…
- … e allora come è possibile la programmazione asincrona?
- Oltre all'interprete JavaScript, nel browser c'è anche un **Web API / Gestore Eventi**

JavaScript nel browser



Live demo

JavaScript nel browser - esempio



Una precisazione su setTimeout

- setTimeout accetta due argomenti:
  - La funzione da eseguire dopo un certo intervallo di tempo
  - L'intervallo di tempo stesso (in millisecondi)
- Trascorso l'intervallo di tempo specificato:
  - Le istruzioni da eseguire sono solo messe in coda
  - Sono eseguite solo quando (e se) quelle già in coda sono terminate
  - Dalla chiamata di setTimeout all'esecuzione delle istruzioni può passare un tempo maggiore di quello specificato

Contatti

Email: paolocaramanica@gmail.com

Linkedin: https://it.linkedin.com/in/paolo-caramanica-436942a/it-it

Facebook: https://it-it.facebook.com/paolo.caramanica

Twitter: https://twitter.com/PaoloCaramanica

GitHub: https://github.com/paolocaramanica

Q&A