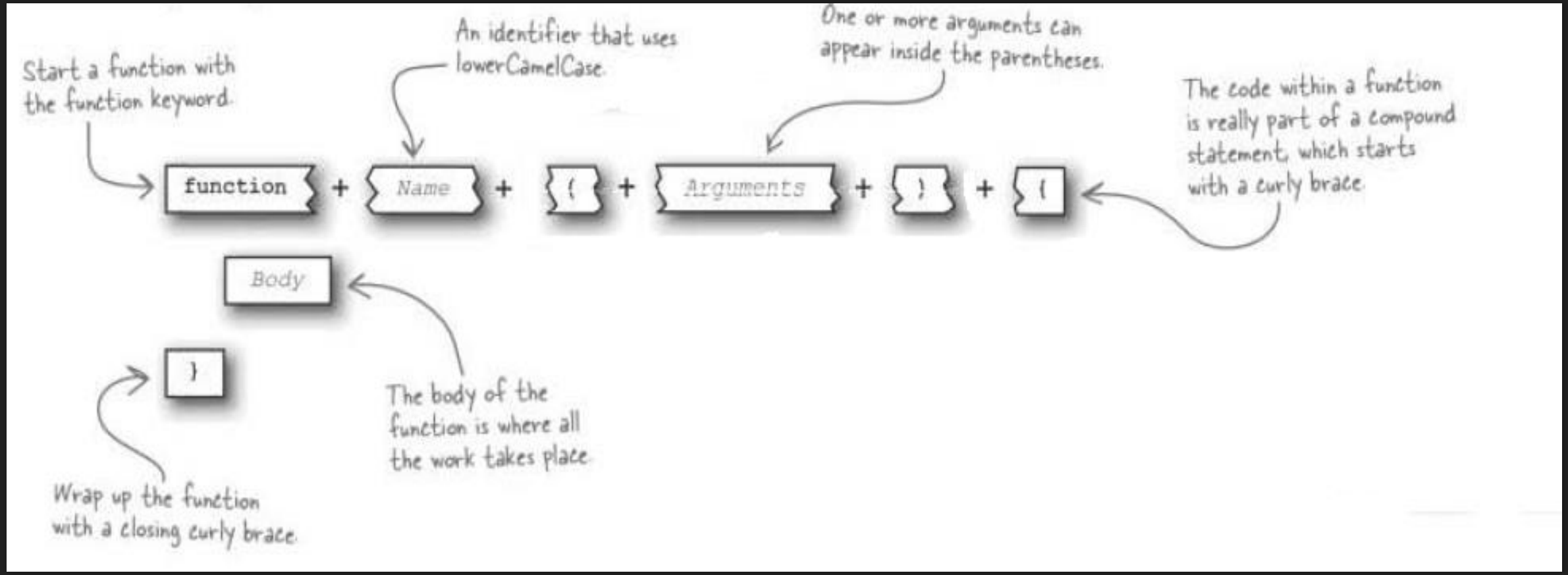
Capitolo 6: le funzioni

Santinelli Marco

Perché usarle?

- Aumentano la leggibilità del codice
- Semplificano il lavoro se si divide il problema principale in sotto-problemi risolvibili tramite una funzione
- Evitano di dover copiare più volte lo stesso pezzo di codice
- Rendono più facile la manutenzione del codice
- Spesso sono già state create da qualcun'altro

Dichiarazione



Return

La return è un istruzione che consente di far restituire alla funzione un certo valore.
Può essere usata in 2 modi:

- return seguito da una variabile o da un'espressione:
  Viene calcolato il valore dell'espressione e viene restituito il valore

```
return valore;
return 3*i-fattoriale(5);
```

- return da solo
  La funzione termina senza restituire nulla

```
return;
```

Continuiamo a migliorare il nostro cinema

Vogliamo:

- Migliorare il vecchio codice usando le funzioni
- Aggiungere nuove funzionalità

Eliminiamo il codice duplicato

Prima

```
for (var i = 0; i < seats.length; i++) {
    for (var j = 0; j < seats[i].length; j++) {
        if (seats[i][j]) {
            // Set the seat to available
            document.getElementById("seat" + (i * seats[i].length + j)).src = "seat_avail.png";
            document.getElementById("seat" + (i * seats[i].length + j)).alt = "Available seat";
        } else {
            // Set the seat to unavailable
            document.getElementById("seat" + (i * seats[i].length + j)).src = "seat_unavail.png";
            document.getElementById("seat" + (i * seats[i].length + j)).alt = "Unavailable seat";
        }
    }
}
```

Dopo

```
function setSeat(seatNum, status, description) {
    document.getElementById("seat" + seatNum).src = "seat_" + status + ".png";
    document.getElementById("seat" + seatNum).alt = description;
}

for (var i = 0; i < seats.length; i++) {
    for (var j = 0; j < seats[i].length; j++) {
        if (seats[i][j]) {
            setSeat(i * seats[i].length + j, "avail", "Available seat");
        } else {
            setSeat(i * seats[i].length + j, "unavail", "Unavailable seat");
        }
    }
}
```

- Nuova funzionalità: un aiuto per i daltonici

```
function getSeatStatus(seatNum) {
  if (selSeat != -1 && (seatNum == selSeat || seatNum == (selSeat + 1) || seatNum == (selSeat + 2)))
    return "yours";
  else if (seats[Math.floor(seatNum / seats[0].length)][seatNum % seats[0].length])
    return "available";
  else
    return "unavailable";
}
```

- Associamo la funzione al click sul sedile

```
<body onload="initSeats();">
  <div style="margin-top:25px; text-align:center">
    <img id="seat0" src="" alt="" onclick="alert('This seat is ' + getSeatStatus(0));" />
    <img id="seat1" src="" alt="" onclick="alert('This seat is ' + getSeatStatus(1));" />
    <img id="seat2" src="" alt="" onclick="alert('This seat is ' + getSeatStatus(2));" />
    <img id="seat3" src="" alt="" onclick="alert('This seat is ' + getSeatStatus(3));" />
    <img id="seat4" src="" alt="" onclick="alert('This seat is ' + getSeatStatus(4));" />
    <img id="seat5" src="" alt="" onclick="alert('This seat is ' + getSeatStatus(5));" />
    <img id="seat6" src="" alt="" onclick="alert('This seat is ' + getSeatStatus(6));" />
                    ..............................
    <img id="seat35" src="" alt="" onclick="alert('This seat is ' + getSeatStatus(35));" /><br />
    <input type="button" id="findseats" value="Find Seats" onclick="findSeats();" />
  </div>
</body>
```

Separiamo contenuto e funzionalità

- HTML e Javascript sono mescolati
- È buona norma tenere separati HTML, CSS e Javascript
- Manutenzione complicata

I nomi delle funzioni sono dei puntatori

- I nomi delle funzioni, sono come variabili che contengono il riferimento alla funzione

```
function showSeatStatus(seatNum) {
    alert("This seat is " + getSeatStatus(seatNum));
}
```

```
var showSeatStatus = function(seatNum) {
    alert("This seat is " + getSeatStatus(seatNum));
}
```

- Le due funzioni sono uguali e infatti si richiamano nello stesso modo

- Posso inoltre dichiarare più puntatori alla stessa funzione

```
var funz = showSeatStatus;
funz(10) mi dà lo stesso risultato di showSeatStatus(10);
```

Collegamento funzione – evento

Tramite HTML

- Il collegamento avviene grazie agli attributi dell'HTML

```
<body onload="initSeats();">
  <div style="margin-top:25px; text-align:center">
    <img id="seat0" src="" alt="" onclick="showSeatStatus(0);" />
    <img id="seat1" src="" alt="" onclick="showSeatStatus(1);" />
```

Tramite riferimenti

- Il collegamento avviene nello script

```
window.onload = initSeat;
```

- NB:

```
window.onload = initSeat;  ≠  window.onload = initSeat();
```

La prima è un riferimento alla funzione, la seconda è una chiamata alla funzione.

Non è possibile collegare direttamente una funzione con dei parametri ad un evento tramite riferimenti

Function literal

Vengono usate quando bisogna collegare delle funzioni con parametri agli eventi.
Dato che non hanno un nome, sono chiamate anche funzioni anonime

```
function(evt) {
                showSeatStatus(0);
               };
```

All'interno della function literal richiamo la funzione con i parametri.
Il parametro evt contiene delle informazioni specifiche per ogni evento

Per collegare una function literal ad un evento basta fare:

```
document.getElementById("seat0").onclick = function(evt) {
                showSeatStatus(0);
            };
```

Oppure:

```
window.onload = function(evt) {
    showSeatStatus(0);
};
```

Uniamo i pezzi

Creiamo una function literal che inizializza la pagina e collega ogni evento alla propria funzione

```
window.onload = function() {
    // Wire the Find Seat button event
    document.getElementById("findseats").onclick = findSeats;

    // Wire the seat image events
    document.getElementById("seat0").onclick = function(evt) { showSeatStatus(0); };
    document.getElementById("seat1").onclick = function(evt) { showSeatStatus(1); };
    document.getElementById("seat2").onclick = function(evt) { showSeatStatus(2); };
    document.getElementById("seat3").onclick = function(evt) { showSeatStatus(3); };
                    ..............................................
    document.getElementById("seat34").onclick = function(evt) { showSeatStatus(34); };
    document.getElementById("seat35").onclick = function(evt) { showSeatStatus(35); };

    // Initialize the seat appearances
    initSeats();
};
```

Funzione senza parametri

Funzioni con parametri

Semplice chiamata di una funzione

Così facendo abbiamo separato il codice Javascript dall'HTML

```
<html>
  <head>
    <title>Mandango - The Macho Movie Ticket Finder</title>

    <script type="text/javascript">
  </head>

  <body>
    <div style="margin-top:25px; text-align:center">
      <img id="seat0" src="" alt="" />
      <img id="seat1" src="" alt="" />
      <img id="seat2" src="" alt="" />
      <img id="seat3" src="" alt="" />
      <img id="seat4" src="" alt="" />
      <img id="seat5" src="" alt="" />
                .............
      <img id="seat30" src="" alt="" />
      <img id="seat31" src="" alt="" />
      <img id="seat32" src="" alt="" />
      <img id="seat33" src="" alt="" />
      <img id="seat34" src="" alt="" />
      <img id="seat35" src="" alt="" /><br />
      <input type="button" id="findseats" value="Find Seats" />
    </div>
  </body>
</html>
```

```
<script type="text/javascript">
  var seats = [[ false, true, false, true, true, true, false, true, false ],
               [ false, true, false, false, true, false, true, true, true ],
               [ true, true, true, true, true, true, false, true, false ],
               [ true, true, true, false, true, false, false, true, false ]];
  var selSeat = -1;

  function initSeats() {

  function getSeatStatus(seatNum) {

  function showSeatStatus(seatNum) {

  function setSeat(seatNum, status, description) {

  function findSeats() {

  // Wire the events
  window.onload = function() {
</script>
```

FINE