Videocorso PHP

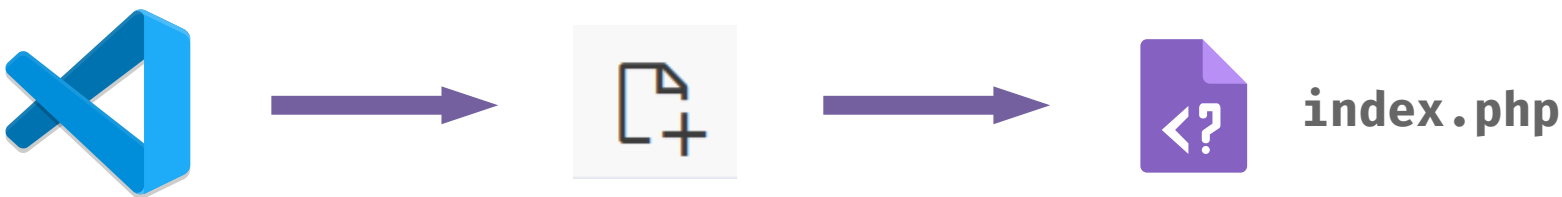
I form e JavaScript

richiami del codice lato client e input PHP

Alessandro Flora

Come sarà impostata questa lezione

- All'interno delle slide ci saranno alcuni riferimenti utili per il successivo ripasso
- In particolare nelle slide sono riportati i concetti di base e la sintassi
- La seconda parte sarà un esercizio svolto; è vivamente consigliato provare a ragionare insieme creando uno script per provare in prima persona



L’input in PHP

Ricevere degli input in PHP

- Per farlo dobbiamo usare una richiesta GET o POST
- Per le richieste GET basta aggiungere il **?** e i parametri dopo di esso
- Per le richieste POST dobbiamo creare un form o usare una richiesta Ajax
- In ogni caso accederemo ai dati sotto forma di stringhe grazie ai vettori associativi **$_GET** e **$_POST**

L'input in PHP

I form HTML

- Sono i moduli: insiemi di campi di testo, elenchi a discesa, selezioni multiple o singole, ...
- Possono inviare i dati inseriti al loro interno mediante una richiesta GET o una richiesta POST a una pagina PHP inserita come valore all'attributo **`action`**
- Il vettore **`$_GET`** o **`$_POST`** avrà per chiavi i valori degli attributi name dei campi del modulo
- Devono terminare con un input di tipo **`submit`** per consentire l'invio del modulo

L'input in PHP

I form HTML

Il metodo può essere GET o POST

Una volta premuto invio si verrà portati alla pagina **elabora.php** tramite una POST con i dati del modulo

```
<form method="POST" action="elabora.php">
    <input type="text" name="campoTesto" required>
    <input type="submit" value="Invia">
</form>
```

Una volta inviato il modulo si otterrà il vettore:

```
$_POST = [
    'campoTesto' => 'valore'
];
```

come input nella pagina **elabora.php**

L'input in PHP

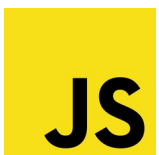
I principali input dei form HTML

```html
<input type="text" placeholder="Legenda"> <!-- campo di testo -->
<textarea rows="numRighe" cols="numColonne"></textarea> <!-- area di testo -->
<input type="password" placeholder="legenda"> <!-- campo password (caratteri nascosti) -->
<input type="color"> <!-- colore (finestra per selezionarlo) -->
<input type="date"> <!-- data (si apre il calendario) -->
<input type="number"> <!-- numero -->
<input type="tel"> <!-- numero di telefono -->
<input type="reset"> <!-- bottone di reset del modulo -->
<input type="time"> <!-- ora (si apre l'orologio) -->
<input type="file"> <!-- file (si apre una finestra per selezionare il file) -->

<label for="casella">Legenda casella</label>
<input type="checkbox" id="casella"> <!-- casella selezionabile -->

<!-- selezione radio (esclusiva di una delle opzioni) -->
<label for="selezioneRadio1">Legenda selezione radio 1</label>
<input type="radio" name="sceltaRadio" id="selezioneRadio1" value="1">
<label for="selezioneRadio2">Legenda selezione radio 2</label>
<input type="radio" name="sceltaRadio" id="selezioneRadio2" value="2">

<!-- menu a discesa -->
<label for="selezionaDiscesa">Legenda discesa</label>
<select id="selezionaDiscesa">
    <optgroup label="Prime due opzioni">
        <option value="1" selected>Opzione 1</option>
        <option value="2" disabled>Opzione disabilitata</option>
    </optgroup>
    <option value="3">Terza opzione</option>
</select>
```

Richiami di JavaScript

Il codice lato client

- Quando sviluppiamo un'applicazione web è sempre bene evitare di caricare il server per spostare le elaborazioni quanto più possibile sul client
- Per codificare degli script lato client (verranno dunque eseguiti sul browser) si usa il linguaggio JavaScript (eventualmente con il framework JQuery)
- Negli ultimi anni Microsoft ha rilasciato TypeScript ovvero una versione di JavaScript fortemente tipizzata che è utile per velocizzare la scrittura del codice e risolve alcune ambiguità del linguaggio



La sintassi di JavaScript

- Per quanto riguarda i costrutti di base è identica a quella vista per PHP (senza i dollari per le variabili e il controllo dei tipi per le funzioni)

```
if(a==b) {
    // mostra un popup nella pagina
    window.alert("Il valore di a è uguale al valore di b");
} else if(a > b) {
    // stampa il messaggio nella console del browser (F12 -> console)
    console.log("Il valore di a è maggiore del valore di b");
}

while(a > b) {
    console.log("Il valore di a è maggiore del valore di b");
    a++;
}

for(i=0;i<a.length;i++) { // la proprietà length del vettore a corrisponde alla dimensione
    window.alert("Il "+i+" elemento del vettore a è "+a[i]);
}

a = [1,2,3,4,5]; // inizializzazione di un vettore

function prova() {
    // funzione di prova
    console.log("La funzione è stata chiamata");
    return 1;
}
```

Il punto e virgola può essere omesso se si va a capo (è comunque meglio metterlo per non disabituarsi negli altri linguaggi)

Gli oggetti JavaScript

- Sono un'implementazione diversa dei vettori associativi che abbiamo visto in PHP
- Sono utili per interfacciarci con PHP usando come tramite il formato JSON

```
// creiamo un oggetto con dei campi
var oggetto = {
    proprieta1: 'valore1',
    proprieta2: 2,
    esempioBooleano: true,
    campoFloat: 2.4
};

// creiamo un oggetto vuoto
var oggetto_vuoto = Object();

// creiamo la chiave campo a cui assegnamo il valore valore1
oggetto_vuoto['campo'] = 'valore1';

// selezioniamo due chiavi dell'oggetto (i metodi sono circa equivalenti)
// se il campo non esiste otteniamo undefined
console.log(oggetto['proprieta1']);
console.log(oggetto.proprieta2);
```

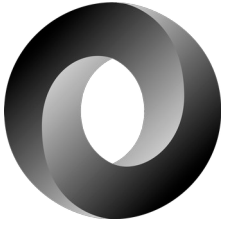
Le variabili (fra cui gli oggetti) vanno inizializzate con la keyword **`var`**

Da oggetti a JSON e viceversa

Possiamo usare le seguenti funzioni che prendono una stringa JSON e la trasformano in un oggetto o viceversa

```
// creiamo un oggetto con dei campi
var oggetto = {
    proprieta1: 'valore1',
    proprieta2: 2,
    esempioBooleano: true,
    campoFloat: 2.4
};

// passiamo dall'oggetto a una stringa JSON
var stringa_json = JSON.stringify(oggetto);

// torniamo dalla stringa JSON a un oggetto JavaScript
var oggetto_da_json = JSON.parse(oggetto);
```

```
{"proprieta1":"valore1","proprieta2":2,"esempioBooleano":true,"campoFloat":2.4}
```



Le Arrow function

- Sono delle funzioni definite *in linea* ovvero senza dover scrivere la funzione prima del suo effettivo uso
- Possono ricevere dei parametri in input e restituirne di altri
- Se definite all'interno di una variabile possono essere richiamate chiamando la variabile come se fosse una funzione

```
// definiamo l'arrow function all'interno di una variabile
var arrowFunction = () => {
    console.log("Hello, World!");
};

// richiamiamo l'arrow function
arrowFunction();
```

Notifiche in JavaScript

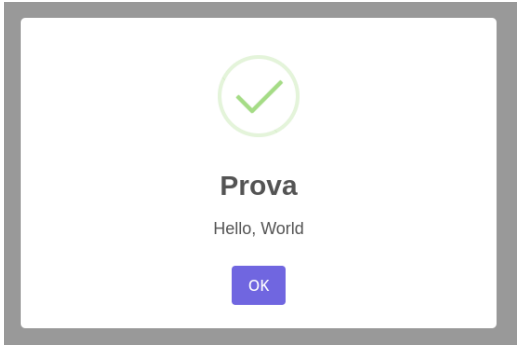
La libreria Swal2

- Può essere utilizzata per inviare notifiche all'utente
- Deve essere collegato lo script JS mediante l'istruzione

```
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11.12.4/dist/sweetalert2.all.min.js"></script>
```

- Per richiamare una swal usiamo l'istruzione:

```
Swal.fire();
```



All'interno delle tonde possiamo passare una stringa o un oggetto (vedi slide successiva)

Notifiche in JavaScript

I parametri a Swal.fire

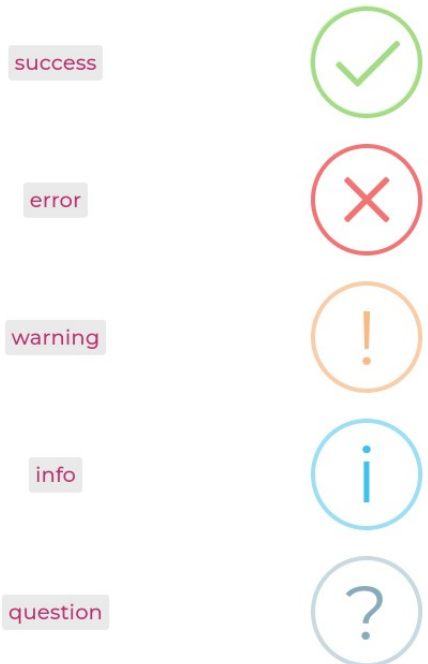
- Li usiamo per personalizzare la notifica
- Li esprimiamo sotto forma di oggetto (o fuori dalla chiamata tramite una variabile o all'interno della chiamata aprendo le tonde e andando a capo)

```
Swal.fire({
    title: 'Titolo della notifica',
    icon: 'nome icona (vedi slide successiva)',
    text: 'testo della notifica',
    html: 'eventuale contenuto html della notifica',
    showConfirmButton: true, // mostra o no il bottone OK
    showCancelButton: true, // mostra o no il bottone Annulla
    showCloseButton: true, // mostra o no la X per chiudere la notifica
    showDenyButton: true, // mostra o no il bottone No
    confirmButtonText: 'testo bottone conferma',
    customClass: { // classi CSS opzionali
        confirmButton: 'classe personalizzata del bottone di conferma',
        denyButton: 'classe personalizzata del bottone di conferma',
    }
});
```

Non è obbligatorio specificarle tutte

Notifiche in JavaScript

Icone delle Swal2



È importante ricordarsi di inserire il nome tra virgolette o tra apici (è una stringa, non una variabile)

Quando vengono chiamate mostrano una animazione di apertura

Notifiche in JavaScript

Comportamento dell'utente

- Possiamo usare una Arrow function a seguito di una Swal usando l'attributo **`then`**
- Se vogliamo determinare il comportamento dell'utente dobbiamo passare un argomento alla Arrow function

```
Swal.fire({
    icon: 'info',
    title: 'Conferma',
    text: 'Vuoi confermare la tua scelta?',
    showConfirmButton: true,
    confirmButtonText: 'Conferma',
    showDenyButton: true,
    denyButtonText: 'No',
    showCancelButton: false,
    showCloseButton: false
}).then((risposta) => { // all'interno del then definiamo la Arrow function
    if(risposta.isConfirmed) {
        console.log("L'utente ha risposto Conferma");
    } else if(risposta.isDenied) {
        console.log("L'utente ha risposto No");
    } else {
        console.log("Comportamento imprevisto");
    }
});
```