**EL3-9\_4ST DATE E TEMPO**

**Autore: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Data: \_\_\_\_\_\_\_\_\_Classe: \_\_\_\_**

**ATTENZIONE**

Il significato dei simboli usati in questa e nelle prossime esercitazioni è spiegato dettagliatamente nella guida <http://www.classiperlo.altervista.org/Materiale/Generale/Simboli.doc> (scaricala e consultala in caso di dubbi).

|  |
| --- |
|  |

Il simbolo della manina in colore blu indica una domanda alla quale bisogna OBBLIGATORIAMENTE rispondere scrivendo in colore blu.

|  |
| --- |
|  |

Il simbolo della manina in colore blu con la scritta Cou New indica un codice che va incollato usando Courier New in colore blu.

|  |
| --- |
|  |

Il simbolo della manina in nero indica una o più immagini o schermate da incollare (protette col tuo watermark, le tue iniziali di Nome e Cognome)



Il simbolo della manina con colori attenuati indica un'operazione che bisogna svolgere, senza rispondere a nessuna domanda (non vuol dire che non devi fare nulla - significa solo che non devi scrivere niente!).



Il simbolo della manina col ciak video indica un video da registrare con *Gif Recorder* (<http://gifrecorder.com/>)e da salvare in formato gif nella cartella dell'esercitazione.

recupero.

E:\Dropbox\Doc\Sito Web\classiperlo\immagini\video.gifQuesto simbolo indica un video di esempio o di spiegazioni da guardare su YouTube

**A) OPERAZIONI PRELIMINARI**

A1) Compila l'intestazione del documento scrivendo Nome, Cognome e Classe



A2) Crea una sottocartella di ES3 con nome uguale a quello di questa esercitazione (*EL3-9\_4ST Date e tempo*)



A3) All'interno della sottocartella *EL3-9\_4ST Date e tempo* salva questo file Word



**B) L'OGGETTO DATE E LE MISURE DI TEMPO**

Il JS contiene molte *funzioni di library* per la misura del tempo.

|  |
| --- |
|  |

B1) Cosa sono le *funzioni di library*? Non copiare una definizione che non capisci da Google!!!

Usare queste funzioni è abbastanza semplice, ma bisogna conoscere un piccolo trucco. Ecco come si fa.

**DICHIARARE E USARE UNA VARIABILE OGGETTO DI TIPO DATE**

Per prima cosa nel tuo programma devi inizializzare una variabile in un modo un po' strano:

*oggettoData* = new Date();

Il nome della variabile è *oggettoData* e ovviamente puoi sostituirlo con quello che ti piace di più.

Tutto il resto nell'istruzione di dichiarazione della variabile *oggettoData* deve invece essere scritto esattamente come ho fatto io (in altri termini: var, new e Date() sono *parole chiave* del linguaggio JS che non devono essere modificate).

Una volta che hai dichiarato una variabile di tipo Date(), puoi usarla per ricavare tutte le informazioni che vuoi sulla data corrente. Per esempio:

*oggettoData*.getFullYear() torna l'anno corrente

*oggettoData*.getHours torna l'ora corrente

Presta molta attenzione al punto (*dot*) che separa il nome della variabile oggetto (nel mio caso *oggettoData*) dal nome della funzione di library (detta più precisamente metodo, es getHours).

Ok, ma veniamo al sodo. Eccoti subito un esempio di come posso fare per usare le funzioni per il tempo (che in realtà si chiamano **metodi**, ma chi se ne importa...) del JS:

data = new Date();

Hh = data.getHours() + ":";

Mm = data.getMinutes() + ":";

Ss = data.getSeconds() + ":";

mm = data.getMilliseconds() ;

gg = data.getDate();

me = data.getMonth()+1;

aa = data.getFullYear();

alert("Sono le ore " + Hh + Mm + Ss + mm + " del giorno "+gg+"/"+me+"/"+aa);

B2) → *tempo1.html (HTML + CSS + JS, insieme)*

Crea una pagina con una funzione di interfaccia chiamata sull'evento onLoad del BODY (caricamento della pagina - se non ricordi come si fa, dai un'occhiata all'esercitazione 3.8). Dentro la funzione incolla il codice qui sopra.

|  |
| --- |
|  |

B3) → Schermata browser

Visto? Se hai fatto tutto correttamente dovresti vedere una finestrella di pop-up nel quale viene visualizzata la data corrente (letta dall'orologio interno del tuo PC).

Non c'è molto da dire sulle varie funzioni per il calcolo del tempo. Per esempio getDate() torna il giorno del mese (es. 28) e getFullYear() ti dice in quale anno siamo.

L'unica cosa da sapere è che, per poter usare questi metodi, devi prima inizializzare un oggetto di tipo Date(). Non importa che nome dai a questo oggetto. Per esempio:

anno = new Date();

anno = anno.getFullYear();

alert("Siamo nel "+pluto);

|  |
| --- |
|  |

B4) Quale scritta viene visualizzata eseguendo l'istruzione *alert* nell'esempio qui sopra? Se mi dici che visualizza la scritta "Siamo nel anno" possiamo pure cambiare mestiere tutti e due (tu e io).

Devi solo stare attento al fatto che data.getMonth() numera i mesi a partire da 0 (gennaio) fino a 11 (dicembre), per cui se vuoi il mese in un formato adatto agli esseri umani, devi sommare uno al risultato:

me = data.getMonth()+1;

Eccoti qui i riferimenti per la spiegazioni e gli esperimenti su *Programmiamo* e *W3School*:

 <http://www.programmiamo.altervista.org/js/date/date1.html>

 <https://www.w3schools.com/jsref/jsref_obj_date.asp>

E:\Dropbox\Doc\Sito Web\classiperlo\immagini\video.gif<https://youtu.be/euVInQ35vrs>

B5) → *tempo2.html (HTML + CSS + JS, insieme)*

Realizza una pagina con un pulsante che funziona come nel video precedente. ATTENZIONE al fatto che il testo deve essere visualizzato dentro a un paragrafo o a un div sotto al pulsante (devi usare innerHTML per scrivere dentro un elemento di blocco).

|  |
| --- |
|  |

B6) → codice *tempo2.html*

**C) LO SFIGOMETRO**

Alcune persone pensano che il venerdì sia il giorno più sfortunato della settimana, il 17 il giorno più sfortunato del mese e gli anni bisestili siano i più sfigati di tutti.

È perciò molto utile avere un programma il quale esamini la data odierna e ti dica quanto è sfigato il giorno corrente (uno sfigometro o misuratore di sfiga, appunto).

**LIVELLO DI SFIGA**

La classificazione del livello della sfiga è la seguente:

- una faccina che ride ☺ se il giorno non è sfigato

- una faccina triste ☹ per le date che cadono di venerdì, di 17 o in un anno bisestile

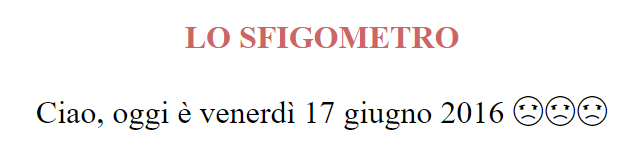
- due faccine tristi ☹☹ se si verificano due delle condizioni di sfiga (es. venerdì 17 oppure un 17 in un anno bisestile)

- tre faccine tristi ☹☹☹ per i venerdì 17 degli anni bisestili (in quei giorni è più prudente restare a casa)

La pagina contiene solo un h1 e un p (oppure un div) e al caricamento appare così\_



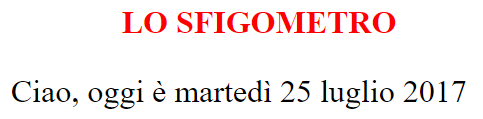
Oppure:



Nota come è cambiato anche il colore dell'intestazione H1.

C1) → *tempo3.html (HTML + CSS + JS, insieme)*

Per prima cosa occupati solo di costruire la pagina e di far visualizzare la data al caricamento, senza preoccuparti né di cambiare il colore del titolo né di disegnare le faccine:



**CONOSCERE IL GIORNO DELLA SETTIMANA**

Per il giorno della settimana (lunedì...domenica) devi usare il metodo getDay:

data = new Date();

gs = data.getDay(); //gs è il giorno della settimana in numero

getDay() torna un numero che corrisponde al giorno della settimana (0 è domenica, 1=lunedì eccetera). A questo punto possiamo convertire il numero in una stringa con una serie di if:

if (gs == 0) { giorno = "Domenica" }

if (gs == 1) { giorno = "Lunedi" }

if (gs == 2) { giorno = "Martedì" }

if (gs == 3) { giorno = "Mercoledì" }

if (gs == 4) { giorno = "Giovedì" }

if (gs == 5) { giorno = "Venerdì" }

if (gs == 6) { giorno = "Sabato" }

Un modo più elegante per realizzare la conversione precedente, consiste nell'usare l'istruzione switch case (<https://www.w3schools.com/js/js_switch.asp>):

switch (gs) {

case 0:

giorno = "Domenica";

break;

case 1:

giorno = "Lunedì";

break;

....

}

C2) Occupiamoci adesso di capire se il giorno è venerdì (gs, giorno della settimana, 5) e/o la data (gg, giorno del mese) è 17 (ci occuperemo poi del problema degli anni bisestili):

if (gg != 17 && gs != 5) {

// h1 in colore1 e una faccina che ride

}

else if (gg == 17 && gs != 5 || gg != 17 && gs == 5) {

// h1 in colore2 e una faccina triste

}

else if (gg == 17 && gs == 5) {

// h1 in colore3 e due faccine tristi

}

**COME FUNZIONA**

Ho usato 3 colori diversi per l'intestazione h1 (indicati con colore1, colore2 e colore3) a seconda del livello di sfiga (scegli liberamente i colori).

Osserva la condizione del secondo else if:

gg == 17 && gs != 5 || gg != 17 && gs == 5

dove && indica AND e || indica OR. Non sono necessarie parentesi perché && ha sempre la precedenza su ||, così come in aritmetica la moltiplicazione viene sempre eseguita prima della somma. Se però, per tua tranquillità, preferisci mettere le parentesi, puoi farlo:

(gg == 17 && gs != 5) || (gg != 17 && gs == 5)

|  |
| --- |
|  |

C3) → codice *tempo3.html (senza anni bisestili, con faccine e colore diverso di h1)*

**PER DISEGNARE LE FACCINE**

Per la faccina che ride usa il carattere &#9786; e per la faccina triste usa &#9785; (codice UTF-8 dei due caratteri). Se vuoi una lista più completa di simboli, guarda qui <http://www.sabinanore.com/design/html-special-symbols/>

Per esempio per disegnare tre faccine tristi devi usare la seguente stringa:

"&#9785; &#9785; &#9785;"

Il problema del programma precedente è come fare per testarne il funzionamento. Infatti provarlo solo nel giorno corrente (oggi) non basta per sapere se il programma funziona correttamente. D'altra parte cambiare la data del PC per provare il programma sarebbe piuttosto scomodo, non ti pare?

E:\Dropbox\Doc\Sito Web\classiperlo\immagini\video.gif<https://youtu.be/HedQxwJraxA>

C4) → copia *tempo3.html* in *tempo4.html* e aggiungi un pulsante e una casella di testo come mostrato nel video

**ATTENZIONE AL FORMATO DELLE DATE**

Come puoi vedere nel video, la data deve essere fornita in formato anglosassone mm/gg/aaaaa (prima il giorno e poi il mese):



Per fortuna c'è una soluzione brillante. Possiamo usare il contenuto di una casella di testo per assegnare all'oggetto Date la data di prova:

dataProva = document.getElementById("dataProva").value;

data = new Date(dataProva);

**COME FUNZIONA**

Se nella dichiarazione della variabile data non metto nulla fra le parentesi

data = new Date();

la data viene presa dall'orologio del PC (data corrente).

Se invece metto fra parentesi una stringa contenente una data valida

data = new Date("03/29/2003");

la data viene inizializzata in base al contenuto della stringa.

A questo punto basta chiamare sull'evento onClick la stessa identica funzione di interfaccia che chiami al caricamento della pagina:

<input type="button" value="PROVA" onclick="calcoloData()">

Dentro alla funzione di interfaccia, un semplice if ti permette di scegliere se usare la data di sistema (quella dell'orologio) oppure quella contenuta nella casella di testo:

dataProva = document.getElementById("dataProva").value;

if (dataProva != "")// se la casella di testo è vuota

data = new Date(dataProva);

else

data = new Date();

Ti faccio osservare due cose importanti, a proposito di questo spezzone di codice:

1) non ho convertito in formato numerico il valore letto da input (usando Number)

2) se l'if e l'else contengono una sola istruzione, le parentesi graffe non sono obbligatorie (anche in questo caso però puoi metterle, se ti fa sentire più sereno).

|  |
| --- |
|  |

C5) Per quale motivo non ho convertito in formato numerico il valore di input?

C6) → *tempo1.gif* (test di funzionamento, tre prove inserendo la tua data di nascita, la data di ieri e la data di venerdì della scorsa settimana)

**D) ANNI BISESTILI**

Il nostro programma è quasi completo ma manca il controllo sull'anno bisestile. Infatti se l'anno è bisestile bisogna aggiungere una faccina in più (sale il livello di sfiga). Possiamo, come al solito, risolvere tutto usando una funzione di elaborazione.

Il bello delle funzioni (di elaborazione) è che puoi copiarle, incollarle e utilizzarle nei tuoi programmi senza dover fare la fatica di capire come funzionano.

Ora ti mostrerò un sito dove ci sono decine e decine di funzioni pronte per essere copiate e incollate. Si chiama **Rosetta CODE**.

**LA STELE DI ROSETTA (ROSETTA STONE)**

Se oggi siamo in grado di leggere i geroglifici egiziani è grazie ad una stele di granite nera, scoperta nel 1799 dal potente esercito francese sbarcato in Egitto al comando di Napoleone Bonaparte.

Fra gli oggetti raccolti durante la spedizione napoleonica c’era questo blocco di granite su cui era incisa una dedica al faraone Tolomeo V Epifore in tre differenti lingue: geroglifico, demotico e in greco. Poiché la pietra fu ritrovata presso la città di Rosetta, sul Nilo, venne chiamata Stele di Rosetta.



D1) Guarda qui: <https://rosettacode.org/wiki/Category:JavaScript>



La funzione per verificare se un anno è bisestile si trova cercando *Leap Year* (anno bisestile in inglese).

**ATTENZIONE**

Rosetta Code ti propone due diverse funzioni per controllare se un anno è bisestile. Entrambe si chiamano isLeapYear, sono funzioni booleane e sono scritte in modo un po' strano!

In JS una funzione può essere scritta come siamo abituati noi:

function isLeapYear(year) {...

oppure in un modo un po' diverso (come proposto da Rosetta Code):

let isLeapYear = function (year){...

Entrambi i modi sono equivalenti e non devi preoccuparti troppo: basta semplicemente copiare e incollare il codice della funzione (una delle due) che ti viene proposta. La chiamata, comunque sia stata scritta la funzione, si fa sempre allo stesso modo:

ris = isLeapYear(anno);

Osserva infine che Rosetta Code non usa la convenzione di mettere un underscore davanti ai nomi dei parametri delle funzioni (es. year invece di \_year). Si tratta infatti solo di una nostra convenzione per distinguere i parametri (interni alla funzione) dagli argomenti (esterni), ma non è una regola del JS!

D2) → copia *tempo4.html* in *tempo5.html* e incolla fra le funzioni anche il codice di isLeapYear copiato da Rosetta Code

**COME SI USA**

isLeapYear è una **funzione booleana** (come tutte quelle che iniziano con is) e torna come risultato true (se l'anno è bisestile) oppure false (se non è bisestile).

La usi passandole l'anno corrente:

anno = data.getFullYear();

bisesto = isLeapYear(anno);

Devi poi risistemare la sequenza degli if...else nel tuo programma per comprendere anche il caso di anno bisestile. Per esempio:

if (gg != 17 && gs != 5 && bisesto==false) {

// h1 in colore1 e faccina che ride

}

else if (gg == 17 && gs != 5 && bisesto==false || gg != 17 && gs == 5 && bisesto==false) {

// h1 in colore2 e una faccina triste

}

else if (gg == 17 && gs == 5 && bisesto==false) {

// h1 in colore3 e due faccine tristi

}

else if (gg == 17 && gs == 5 && bisesto==true) {

// h1 in colore4 e tre faccine tristi

}

...

La sequenza qui sopra è incompleta. Devi considerare tutti i casi possibili e ci vogliono parecchi if...else.

D3) → *tempo2.gif*

Devi fare un test completo di funzionamento del programma inserendo tutte le seguenti date di prova:

* 07/17/2020: venerdì 17 anno bisestile (tre faccine tristi)
* 07/16/2020: giovedì 16 anno bisestile (una faccina triste)
* 07/16/2019: martedì 16 anno non bisestile (una faccina che ride)
* 05/17/2019: venerdì 17 anno non bisestile (due faccine tristi)
* 03/17/2019: domenica 17 anno non bisestile (una faccina triste)

Se per caso qualcosa non funziona correttamente, cerca l'errore nel tuo programma e correggilo!

**E) MISURARE IL TEMPO TRASCORSO**

Fra i metodi dell'oggetto Date ce n'è uno davvero utilissimo per fare misure di tempo. Si chiama now()e torna il numero di millisecondi trascorsi dalle 00.00 del primo gennaio 1970. Il suo utilizzo è semplicissimo. Per esempio:

tempo1 = Date.now(); /in Tempo1 scrive ms trascorsi dal 1/1/1970

**UNIX TIME**

Perché proprio il primo gennaio del 1970? Perché questa data fu scelta in modo arbitrario come data di riferimento (anno zero) per i computer con sistema operativo Unix e da allora è rimasta in uso come riferimento temporale (in fondo anche noi contiamo gli anni da una data arbitraria, no?).

Ma cosa mai ci può servire sapere quanti secondi sono trascorsi dal primo gennaio 1970? Se proprio muori dalla voglia di saperlo qui trovi un orologio che ti mostra l'orario Unix in tempo reale: <https://www.epochconverter.com/clock>

Il motivo per cui a noi interessa così tanto misurare il numero di millisecondi trascorsi a partire dal 1/1/1970 è che in questo modo possiamo facilmente calcolare quanto tempo passa fra due eventi. Ti mostro subito come si fa.

E1) → *tempo6.html (HTML + CSS + JS, insieme)*

Pagina con due pulsanti AVVIA e STOP.

E:\Dropbox\Doc\Sito Web\classiperlo\immagini\video.gif<https://youtu.be/pZks8it8mok>

Cliccando sul pulsante AVVIA viene chiamata una funzione che legge il tempo Unix e lo memorizza in una variabile (es. Tempo1). Cliccando il pulsante STOP viene chiamata una seconda funzione che legge di nuovo il tempo Unix, lo memorizza in una seconda variabile (es. Tempo2) e poi calcola e visualizza la differenza Tempo2-Tempo1.

In questo modo per esempio possiamo costruire un gioco rudimentale per misurare i nostri tempi di reazione.

|  |
| --- |
|  |

E2) → Codice *tempo6.html*

A proposito sempre di tempi di reazione, ecco qualcosa di più complicato...

E3) → *tempo7.html (HTML + CSS + JS, insieme)*

Vogliamo misurare i tempi di reazione con una pagina contenente due caselle di testo e un pulsante AVVIO.

E:\Dropbox\Doc\Sito Web\classiperlo\immagini\video.gif<https://youtu.be/ewsBph7CO4o>

Premendo su AVVIO viene generato un numero casuale a 5 cifre (funzione caso) e questo numero viene visualizzato in una delle due caselle di testo.

L’utente deve scrivere esattamente lo stesso numero nella seconda casella di testo. Quando il numero scritto è uguale a quello casuale visualizzato, il programma visualizza il numero di secondi trascorsi.

**SUGGERIMENTI**:

Per chiamare la funzione di controllo (che verifica se il numero inserito è uguale a quello generato casualmente), usa l’evento onKeyUp (quando viene rilasciato un tasto) sulla casella di testo contenente il numero scritto dall’utente:

<input type="number" id="numero" onKeyUp="stop()">

Se il numero digitato è uguale a quello generato casualmente, viene calcolato il tempo e viene visualizzato con un alert; altrimenti la funzione di controllo non visualizza nulla.

Per ricavare i secondi dai millisecondi basta dividere per 1000.

|  |
| --- |
|  |

E4) → Codice *tempo7.html*

E5) → *tempo3.gif* (test di funzionamento del programma)



**F) OPERAZIONI FINALI**

F1) Controlla di aver risposto a tutte le domande e incollato tutte le schermate. Tutte le caselline dovrebbero avere un segno X, per indicare che hai risposto 



F2) Comprimi le immagini contenute in questo file Word (seleziona un'immagine, scheda *Formato* e poi *Comprimi immagini* e infine *Applica a tutte le immagini del documento*) in modo da ridurne le dimensioni.



F3) Controlla che la cartella di questa esercitazione contenga i seguenti file con i nomi qui indicati:



|  |  |  |
| --- | --- | --- |
| **Nome del file** | **Tipo del file** | **Descrizione** |
| *EL3-9\_4ST Date e tempo* | Word | Il file di questa esercitazione |
| *tempo1.html* | HTML+CSS+JS |  |
| *tempo2.html* | HTML+CSS+JS |  |
| *tempo3.html* | HTML+CSS+JS |  |
| *tempo4.html* | HTML+CSS+JS |  |
| *tempo5.html* | HTML+CSS+JS |  |
| *tempo6.html* | HTML+CSS+JS |  |
| *tempo.html* | HTML+CSS+JS |  |
| *tempo1.gif* | GIF |  |
| *tempo2.gif* | GIF |  |
| *tempo3.gif* | GIF |  |

F4) Chiudi tutti i file, zippa la cartella di questa esercitazione e inviala all'insegnante su Classiperlo.