**Guión de videoconferencia sobre validación de formularios y expresiones regulares**

* Seguimos con la unidad 5, sobre gestión de formularios. Hemos visto hasta ahora:
  + Qué es un formulario.
  + Cómo se accede a los objetos formularios, y sus métodos y propiedades.
  + Componentes de los formularios, y cómo se accede a ellos.
  + Gestión de los elementos de un formulario:
    - Input tipo text.
    - Input tipo checkbox.
    - Input tipo radio-button.
    - Input tipo select
  + Envío y validación de formularios:
  + Expresiones regulares.
* Transparencia 19:
  + ¿Qué es la validación?
  + ¿Dónde se ha de realizar la validación?
* Transparencia 20: ¿En qué consiste la validación?:
* Transparencia 21: Estrucutra lógica de validació en javascript. Uso de expresiones regulares.
* Transparencia 22:
  + ¿Qué es una expresión regular?
  + 2 formas de definir una expresión regular.
  + Ejemplo en consola.
* Transparencia 23: Partes de una expresión regular.
  + Patrón de búsqueda.
  + Modificador:
    - g: ejemplo de w3schools.
    - i: ejemplo en consola.
    - M: ejemplo de textarea para ^ y para $
* Transparencia 24: Clases de caracteres (de todos ellos explicar y ejemplo en consola)
  + .
  + \w y \W
  + \n, \r \t
  + \s
  + \S
  + \d y \D
  + \
* Transparencia 25: Cuantificadores (de todos ellos explicar y ejemplo en consola)
  + Ejercicio: DNI con 8 números seguidos de manera opcional de un – y seguido de una letra mayúscula.
  + Ejercicio: DNI con 8 números seguidos de manera opcional de un – o más y seguido de una letra mayúscula.
  + Ejercicio: DNI con 8 números seguido de un guión o más y seguido de una letra mayúscula.
  + Ejercicio: buscar la cadena a seguida de t de 1 a 2 veces
* Transparencia 26 y 27: Aserciones:
  + Transparencia 26 : Explicar ^: Ejercicio: Buscar una cadena al comienzo de palabra
  + Transparencia 26 : Explicar $: Ejercicio: Buscar una cadena al final de palabra
  + Transparencia 26 : Explicar \b:
    - Ejercicio: buscar una cadena a comienzo de palabra.
    - Ejercicio: buscar una cadena al final de palabra.
  + Buscar palabras inglesas de cualquier longitud, y de 3 caracteres
  + Buscar palabras españolas de cualquier longitud, y de 6 a 9 caracteres
  + Buscar palabras francesas de cualquier longitud, y de 5 caracteres
  + Transparencia 27: Explicar \B. Ejercicio: buscar una cadena en medio de palabra
  + Transparencia 27: Ejercicio: buscar una cadena por la que no comience una palabra
  + Transparencia 27: Ejercicio: buscar una cadena por la que no termine una palabra
  + Transparencia 28: Explicar buscar una cadena solo si está seguida de otra.
    - Hacer ejemplo.
    - Ejercicio: buscar en la cadena “Hay 3 excursiones para 1ESO, 2ESO, 3ESO, 4ESO y 1BAC. Vienen los grupos 1ESO-A, 1BAC-B, 2ESO-C y 3ESO-D”. Tienes que buscar los niveles de la ESO que irán a la excursión. La expresión que soluciona esto sería /\b\d(?=ESO\-[A-Z]\b)/g. Observa que solo devuelve el número, el resto es una condición. Y observa que el 3 seguido de “ excursiones” o de “ESO” o “BAC” no lo devuelve.
    - Buscar en un texto del tipo dato1=valor1, dato2=valor2, dato3=valor3 todos los nombres de los datos, es decir, dato1, dato2, dato3, .... Solución /\w+(?=\=\w+)/gi Siempre y cuando dato1, dato2, dato3, no contengan otros caracteres diferentes a (el alfabeto latino básico, números y guión bajo) Si por ejemplo los caracteres que compongan dato1, dato2, etc pudieran ser del alfabeto español pondríamos /[\wñáéíóúü]+(?=\=[\wñáéíóúü]+)/gi
  + Transparencia 29: Explicar buscar una cadena solo si no está seguida de otra.
    - Ejercicio: buscar cadenas del tipo numero seguido de la cadena ESO si no están seguidas de un guión y una letra mayúscula. Observa que solo devuelve el número seguido de la cadena ESO, el resto es una condición. Utilizar el mismo ejemplo de ESO.
  + Transparencia 29: Explicar buscar una cadena solo si está precedida de otra
    - Buscar en un texto del tipo dato1=valor1, dato2=valor2, dato3=valor3 todos los valores. Solución (?<=\w+\=)\w+ Siempre y cuando dato1, dato2, dato3, valor1, valor2, valor3 … no contengan otros caracteres diferentes a (el alfabeto latino básico, números y guión bajo). Si por ejemplo los caracteres que compongan dato1, dato2, etc pudieran ser del alfabeto español pondríamos.
  + Transparencia 29: Explicar buscar una cadena solo si no está precedida de otra
    - Buscar en la cadena siguiente: “Luis Gómez, Pedro Martín, Juan Jiménez, María Martín, Luisa Martín”, los apellidos Martín que no están precedidos de Maria: /(?<!María)Martín/g
  + Transparencia 30. Grupos, rangos y |:
    - Explicar []:
      * Explicar y hacer ejemplo.
    - Explicar [^]: explicar y hacer ejemplo
    - Explicar rangos y ejemplo.
    - Explicar (), y hacer ambos ejemplos.
  + Transparencia 31. Métodos de RegExp
    - Exec: explicar y hacer ejemplo:
    - Test: explicar y hacer ejemplo
  + Transparencia 32. Métodos de string que utilizan expresiones regulares
    - Match. Lo hemos utilizado en todos los ejemplos
    - Replace. Explicar. La cadena no se modifica, sino que se devuelve el resultado de reemplazar.
      * Ejemplo: Busca en la cadena “Luis Gómez, Pedro Martín, Juan Jiménez, María Martín, Luisa Martín”, los apellidos Martín que no están precedidos de María y sustituyelo por Martínez. cadena.replace(/(?<!María\s)Martín/g,"Martínez")

cadena.replace(/(?<!María)\sMartín/g," Martínez")

* + - Search. Explicar y en el ejemplo anterior ejecutar: cad.search(/(?<!María)Martín/g) Observa qué posición devuelve.
  + Transparencia 33: uso de RegExp.
    - Explicar.
    - Hacer ejemplo
    - Demostrar que no se podría hacer con expresiones literales.
  + Ejercicios:
    - busca números octales: /\b[0-7]+\b/g
    - busca números binarios: /\b[01]+\b/g
    - busca números hexadecimales: /\b[\dA-F]+\b/g
    - Filtrar nombre, apellidos del alfabeto español y que no permite números ni caracteres de puntuación: /^([a-záéíóúüñ]+ ?)+$/i
    - Filtrar día 1-31: /^([12][0-9]|3[01]|[1-9])$/
    - Filtrar número internacional: /^(00|\+)\d{1,3}\s\d{9}$/
    - Filtrar email:

/^([\w\-]+(\.[\w\-]+)[\*@[\w\-]+(\.[\w\-]+)\*\.[\w\-]{2,10})$/](mailto:*@[\w\-%5d+(\.%5b\w\-%5d+)*\.%5b\w\-%5d%7b2,10%7d)$/)

* + - Número decimal (con 2 decimales) positivo o negativo:

/^[\-\+]?\d+[\.\,]\d{2}$/

* + - Fecha correcta:

/^((((0[1-9]|[12]\d|3[01])\/(0[13578]|1[02]))|((0[1-9]|[12]\d|30)\/(0[469]|11))|((0[1-9]|[12]\d)\/02))\/([1-9]\d{3}|0[1-9]\d{2}|00[1-9]\d|000[1-9]))$/

* + - Buscar cadenas:
      * Cadenas que terminan en n ó s: /[\wáéíóúün]+[ns]\b/gi
      * Colores en singular: /\b(roj[oa]|verde|azul|negr[oa]|blanc[oa])\b/gi
      * Buscar el valor de dato2: /\bdato2\=\w+\b/
      * Buscar números de grupos de ESO:

/\b[1-9]ESO(?=\-[A-Z]\b)/g

* + - * Calcular números grupos de 1ESO:

/\b1ESO(?=\-[A-Z]\b)/g