



Examen

JavaScript

## Programando con JavaScript

Construyendo una aplicación…

Tenemos esta estructura visual de la aplicación que vamos a construir gracias a las plantillas:

* Bigpicture
* Deadstocker
* Enigma
* Kelly
* Yourtrip

En base a una de ellas, desarrollaremos una aplicación en la que podremos comprar productos de lujo. Nuestro cliente nos exige ciertas pautas:

* Se utilizará la marca Luxe (imagen: logo\_luxe.png)
* Categorías de productos a ofrecer (cada imagen tiene el nombre de la categoría y subcategoría):
  + Vehículos:
    - Motocicletas
    - Coches
    - Aviación
    - Embarcaciones
  + Personal:
    - Cosmética y cuidado personal
    - Joyas
    - Turismo y viajes
    - Moda
  + Varios
    - Tecnología
    - Arte
    - Restauración
    - Eventos
  + Deportes:
    - Deportes extremos
    - Personal Trainer
* Teniendo en cuenta la organización de las categorías y subcategorías, debemos mostrar los elementos que corresponda en cada momento, es decir:
  + En la primera pantalla que mostremos, sólo habrá 4 categorías (4 imágenes representativas)
  + En la segunda pantalla (después de seleccionar una categoría) mostraremos tantas imágenes como subcategorías tengamos creadas.
* Una vez seleccionemos el producto que deseemos, deberán aparecer un cuadro en la parte inferior de la imagen (ejemplo en los archivos de ventanas modales 2 y 3 cookies) que nos mostrará 2 opciones:
  + Reservar 🡪 nos aparecerá un calendario para poder seleccionar la fecha de reserva. El calendario deberá aparecer en una ventana modal.
  + Comprar online 🡪 mostrará el precio en una ventana modal y un botón que nos permitiría (en un caso hipotético) ir a la pasarela de pago). En todo caso, debería abrir una ventana independiente a la que estemos.

| **Caracter** | **Significado** |
| --- | --- |
| **[\](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-backslash)** | Buscará coincidencias conforme a las siguientes reglas:   * una barra invertida precediendo un caracter simple indica que éste debe ser interpretado como un caracter especial y no de forma literal. * una barra invertida que precede a un caracter especial indica que éste deberá ser interpretado literalmente, esto es, como un caracter simple y no como un caracter especial. |
| **[^](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-caret)** | Coincide con el principio de la entrada. El caracter '^' tiene un significado diferente cuando aparece como el primer caracter en un patrón. |
| **[$](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-dollar)** | Busca el final de la entrada. |
| **[\*](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-asterisk)** | Busca el caracter precedente 0 (cero) o más veces. Es equivalente a {0,}. |
| **[+](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-plus)** | Busca el caracter precedente 1 o más veces. Es equivalente a {1,}. |
| **[?](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-questionmark)** | Busca el caracter precedente 0 (cero) o 1 (una) vez. Es equivalente a {0,1}. Si se utiliza inmediatamente después que cualquiera de los cuantificadores \*, +, ?, o {}, hace que el cuantificador no sea expansivo (encontrando la menor cantidad posible de caracteres), en comparación con el valor predeterminado, que sí es expansivo (encontrando tantos caracteres como le sea posible).  También se utiliza en coincidencias previsivas, como se describe en las entradas x(?=y)  y  x(?!y) de esta tabla. |
| **[.](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-dot)** | (El punto decimal) coincide con cualquier caracter excepto un caracter de nueva linea. |
| **[(x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-capturing-parentheses)** | Busca 'x' y recuerda la busqueda, como el siguiente ejemplo lo muestra. Los parentesis son llamados parentesis de captura. |
| **[(?:x)](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-non-capturing-parentheses)** | Coincide con 'x' pero no recuerda la coincidencia. Los paréntesis son llamados paréntesis no capturadores, y permiten definir subexpresiones para manipular con los operadores. |
| **[x(?=y)](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-lookahead)** | Coincide con 'x' sólo si 'x' es seguida por 'y'. Esto se denomina previsión (lookahead, mirar adelante). |
| **[x(?!y)](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-negated-look-ahead)** | Coincide con 'x' solo si 'x' no es seguida por 'y'. Es una previsión negativa. |
| **[x|y](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-or)** | Coincide con 'x' o 'y'. |
| **[{n}](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-quantifier)** | Coincide exactamente con n ocurrencias de la expreción. N debe ser un entero positivo. |
| **[{n,m}](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-quantifier-range)** | Donde n y m son enteros positivos y n <= m. Coincide con al menos n y no más de m ocurrencias de la expreción. Si se omite m, no tiene límite de máximo. |
| **[[xyz]](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-character-set)** | Grupo de caracteres. Este tipo de patrón coincide con cada caracter dentro de los corchetes, incluyendo [secuencias de escapado](https://developer.mozilla.org/en-US/docs/JavaScript/Guide/Values,_variables,_and_literals#Unicode_escape_sequences). Caracteres especiales como el punto (.) y el asterisco (\*) no son especiales en un grupo, así que no necesitan ser escapados. Puede especificar un rango utilizando un guión, como en el siguiente ejemplo.  El patrón [a-d], que equivale a [abcd], coincide con la 'b' en "brisket" y la 'c' in "city". El patrón /[a-z.]+/ y /[\w.]+/ coinciden con toda la cadena "test.i.ng". |
| **[[^xyz]](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-negated-character-set)** | Grupo de caracteres negativo. Significa que coincide con cualquier cosa que no esté en los corchetes. Puede especificar rangos. Todo lo que funciona en el grupo de caracteres positivo funciona también aquí.  Por ejemplo, [^abc] es lo mismo que [^a-c], y coincide con la 'r' en "brisket" y 'h' en "chop." |
| **[[\b]](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-backspace)** | Coincide con backspace (U+0008). Debe ir entre corchetes. (No confundir con \b.) |
| **[\b](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-word-boundary)** | Coincide con un *limite de palabra*. Un *limite de palabra* coincide con la posicion entre donde un caracter de palabra no viene precedido o seguido por otro. Notese que el límite no estará incluido en la coincidencia. En otras palabras, la longitud del limite es cero. (No confundir con [\b].)  Ejemplos: /\bm/ coincide con la 'm' de "moon" ; /oo\b/ no tiene coincidencias en "moon", porque las 'oo' estan seguidas de una 'n' que es un caracter de palabra; /oon\b/ coincide con 'oon' en "moon", porque 'oon' es el final de la cadena, por lo cual no va seguido de un caracter de palabra; /\w\b\w/ no coincidirá con nada, porque un caracter de palabra no puede estar seguido por ambos, un limite y un caracter de palabra. |
| **[\B](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-non-word-boundary)** | Coincide con un no-limite de palabra. Esto coincide con una posicion donde el anterior y el siguiente caracter son del mismo tipo: ambos son o no son caracteres de palabra. El inicio y el final de una cadena se consideran no palabras. |
| **[\c](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-control)*[X](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-control)*** | Donde *X* es un caracter entre A y Z. Coincide con un caracter de control en un string. |
| **[\d](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-digit)** | Coincide con un caracter de número. Equivalente a [0-9]. |
| **[\D](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-non-digit)** | Coincide con cualquier caracter no numérico. Equivalente a [^0-9]. |
| **[\f](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-form-feed)** | Coincide con un form feed (salto de página) (U+000C). |
| **[\n](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-line-feed)** | Coincide con un line feed (salto de linea) (U+000A). |
| **[\r](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-carriage-return)** | Coincide con un carriage return (retorno de carro) (U+000D). |
| **[\s](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-white-space)** | Coincide con un *caracter de espacio*, entreellos incluidos espacio, tab, salto de página, salto de linea y retorno de carro. |
| **[\S](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-non-white-space)** | Coincide con todo menos caracteres de espacio. |
| **[\t](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-tab)** | Coincide con tab (U+0009). |
| **[\v](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-vertical-tab)** | Coincide con tab vertical (U+000B). |
| **[\w](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-word)** | Coincide con cualquier caracter alfanumérico, incluyendo el guión bajo. Equivalente a [A-Za-z0-9\_]. |
| **[\W](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-non-word)** | Coincide con todo menos *caracteres de palabra*. Equivalente a [^A-Za-z0-9\_]. |
| **[\](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-backreference)*[n](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-backreference)*** | Cuando *n* es un entero positivo, es una referencia hacia alguna subcadena de parentesis dentro de la misma expresion que coincida con el número (contando los parentesis izquierdos). |
| **[\0](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-null)** | Coincide con el caracter NULL (U+0000). No preseda este por otro número, ya que \0<numero> se considera una secuencia octal [escapada](https://developer.mozilla.org/en-US/docs/JavaScript/Guide/Values,_variables,_and_literals#Unicode_escape_sequences). |
| **[\xhh](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-hex-escape)** | Coincide con un caracter en exadecimal hh (dos dígitos hexadecimales) |
| **[\uhhhh](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions" \l "special-unicode-escape)** | Coincide con un caracter unicode con el código hhhh (cuatro dígitos hexadecimales). |