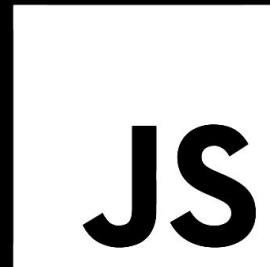


**MIND
HUB.**



Ejercicios módulo Javascript

Condicionales

```
... = modifier_ob...  
... mirror object to mirror...  
... mirror_mod.mirror_object...  
... operation == "MIRROR_X":  
... mirror_mod.use_x = True  
... mirror_mod.use_y = False  
... mirror_mod.use_z = False  
... operation == "MIRROR_Y":  
... mirror_mod.use_x = False  
... mirror_mod.use_y = True  
... mirror_mod.use_z = False  
... operation == "MIRROR_Z":  
... mirror_mod.use_x = False  
... mirror_mod.use_y = False  
... mirror_mod.use_z = True
```

```
... selection at the end -ad...  
... _ob.select= 1  
... _ob.select=1  
... context.scene.objects.active  
... ("Selected" + str(modifier...  
... _ob.select = 0  
... bpy.context.selected_ob...  
... ata.objects[one.name].sel...
```

```
... Int("please select exact...
```

... OPERATOR CLASSES ...

```
... types.Operator):  
... X mirror to the selected  
... object.mirror_mirror_x"  
... or X"
```

**MIND
HUB.**

Resumen Condicionales

- Palabra reservada `"if"` para crear una sentencia de decisión.
- Abrir paréntesis (`expresión a evaluar`) dónde irá la sentencia a evaluar la cual dará siempre tendrá como valor verdadero (`true`) o falso (`false`).
- Abrir llaves `{... bloque de código...}` para escribir la o las acciones a realizar en caso de que la expresión a evaluar haya sido verdadera (`true`).
- Luego de la llave de cierre de la expresión `"if"` se puede utilizar `"else"` por si se quiere realizar alguna acción o acciones cuando la expresión evaluada dentro del `"if"` haya sido falsa (`false`).
- En caso de usar `"else"` se debe abrir nuevamente llaves `{...bloque de código...}` para realizar la o las acciones en el mismo
- También se puede escribir `"else if {...bloque de código...}"` para realizar otra pregunta en caso de haber sido falsa (`false`) el valor del `"if"` inicial y se puede seguir agregando tantos `"else {}"` o `"else if {}"` como se requieran al finalizar una sentencia `"if"` o `"else if"`.

Resumen Condicionales

- Palabra reservada **"switch"** para crear una sentencia de decisión múltiple.
- Abrir paréntesis (variable de control) para colocar la variable que se va a evaluar el valor que contiene.
- Abrir llaves para escribir el bloque de código del **"switch(variable de control){...bloque de código...}"**
- Dentro de las llaves se definen los casos literales a evaluar: **"case 'valor':{...}"**
- Dentro de cada caso se debe usar la palabra reservada **"break"** para finalizar ese caso de uso
- Al final de los casos declarar un caso **"default:{...}"** para cualquier caso no contemplado. - Dato extra, pueden utilizar la función **"confirm('texto de pregunta al usuario')"** para poder preguntar algo por pantalla y según sea la respuesta escogida por el usuario la función devuelve verdadero (**true**) o falso (**false**) según corresponda.

Ejercicios Condicionales

Nº1 | Realizar un programa que dado 2 números imprima por consola si el primer número es mayor que el segundo.

Nº2 | Realizar un programa que dado 2 números imprima por consola si los números son iguales o si son diferentes.

Nº3 | Realizar un programa que dado 2 números imprima por consola cuál de los 2 números es el más grande o si son iguales.

Nº4 | Realizar un programa que dado 3 números imprima por pantalla cual de los 3 es el más chico.

Nº5 | Realizar un programa que dado 2 objetos representando personas con las propiedades nombre, edad y altura imprima por pantalla cual de las 2 personas es la más alta y cual es la de mayor edad.

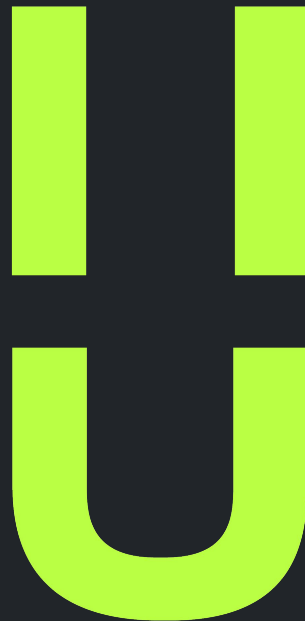
Nº6 | Realizar un programa que permita ingresar por pantalla tu nombre, edad, altura, visión y permita determinar si estás capacitado para conducir. La persona deberá cumplir con una edad mínima de 18 años, medir más de 150 cm y tener una visión de 8 de 10 como mínimo.



Ejercicios Condicionales

Nº7 | Realizar un programa que permita el ingreso de los siguientes datos: Nombre, pase (vip o normal), si posee entrada (si o no, s o n, true o false). Las personas que posean tu mismo nombre tienen ingreso libre así como también los que posean un pase vip, mostrar un mensaje de bienvenida. Si posee entrada preguntar si desea utilizarla, en caso afirmativo mostrar mismo mensaje de bienvenida. Y por último de no tener el mismo nombre o poseer un pase vip o entrada, preguntar si desea comprar, caso afirmativo solicitar dinero disponible, si posee \$1000 o mas, mostrar mensaje de venta de entrada y bienvenida, caso contrario mostrar mensaje de rechazo de venta, y en caso contrario a no querer comprar, mostrar mensaje de despedida.

Nº8 | Realizar un juego de adivinanza estableciendo un valor entre 1 y 10 en una variable llamada **numeroIncognita** y que permita en 3 intentos adivinar el número. El usuario deberá ingresar un numero del 1 al 10 por pantalla en 3 intentos el cual se deberá guardar en una variable llamada **numeroIngresado**, y en cada intento deberás mostrarle un mensaje al usuario diciendo: "el numero ingresado es mayor, vuelve a intentarlo" o "el numero ingresado es menor, vuelve a intentarlo" o en caso de adivinar, un mensaje que diga: "Ganaste, has adivinado el número." No te preocupes si usas mucho código repetido, más adelante verás cómo realizar este juego de manera más eficiente.



Ejercicios Condicionales

Nº9 | Crear un programa que permita ingresar tu edad y decir si eres un infante (0 a 12 años), adolescente (13 a 18 años), un mayor joven (19 a 45 años) o un anciano (más de 45 años), y en caso de ingresar una edad mayor a 100 mostrar un mensaje preguntando si en realidad tiene esa edad.

Nº10 | Crear un programa que permita el ingreso de "PIEDRA", "PAPEL" o "TIJERAS" a 2 jugadores y muestre en pantalla cuál de los 2 ha ganado o si han empatado. En caso de algún ingreso incorrecto mostrar por pantalla que uno de los jugadores ha hecho trampa.

Nº11 | Realizar un programa que permita el ingreso de un color y utilizando "switch" mostrar por pantalla los siguientes mensajes según las opciones:

- Blanco o Negro: Falta de color.
- Verde: El color de la naturaleza.
- Azul: El color del agua.
- Amarillo: El color del sol.
- Rojo: El color del fuego.
- Marrón: el color de la tierra.
- y para cualquier otro valor: Excelente elección, no lo teníamos pensado.



Ejercicios Condicionales

Nº12 | Realizar un programa que permita el ingreso de 2 valores numéricos y una operación. Según sea la operación ingresada (suma, resta, multiplicación, división) el programa deberá mostrar en pantalla un mensaje junto con el resultado. En caso de haber elegido división realizar la operación siempre que sea posible o mostrar un mensaje de ERROR si el divisor ingresado fue 0.

Nº13 | Crear un programa que permita ingresar todos los datos de tu documento nacional de identidad, mostrar por pantalla un mensaje que imprima todos los datos ingresados y pregunte si están correctos los mismos. En caso afirmativo, crear un objeto llamado dni con todos los datos ingresados y mostrarlos por consola con `console.table()` más un mensaje de registro exitoso, en caso de que la persona rechace confirmar los datos, mostrar un mensaje que diga: vuelva a intentarlo en 1 mes.



¡Buena Suerte!

**MIND
HUB.**

```
... = modifier_ob.  
... mirror object to mirror  
... mirror_mod.mirror_object  
... operation == "MIRROR_X":  
... mirror_mod.use_x = True  
... mirror_mod.use_y = False  
... mirror_mod.use_z = False  
... operation == "MIRROR_Y":  
... mirror_mod.use_x = False  
... mirror_mod.use_y = True  
... mirror_mod.use_z = False  
... operation == "MIRROR_Z":  
... mirror_mod.use_x = False  
... mirror_mod.use_y = False  
... mirror_mod.use_z = True
```

```
... selection at the end -ad  
... mirror_ob.select= 1  
... mirror_ob.select=1  
... context.scene.objects.acti  
... ("Selected" + str(modifier  
... mirror_ob.select= 0  
... bpy.context.selected_ob  
... data.objects[one.name].sel  
... print("please select exact
```

OPERATOR CLASSES

```
... types.Operator):  
... an X mirror to the selected  
... object.mirror_mirror_x"  
... mirror X"
```

```
... text):  
... object is not
```