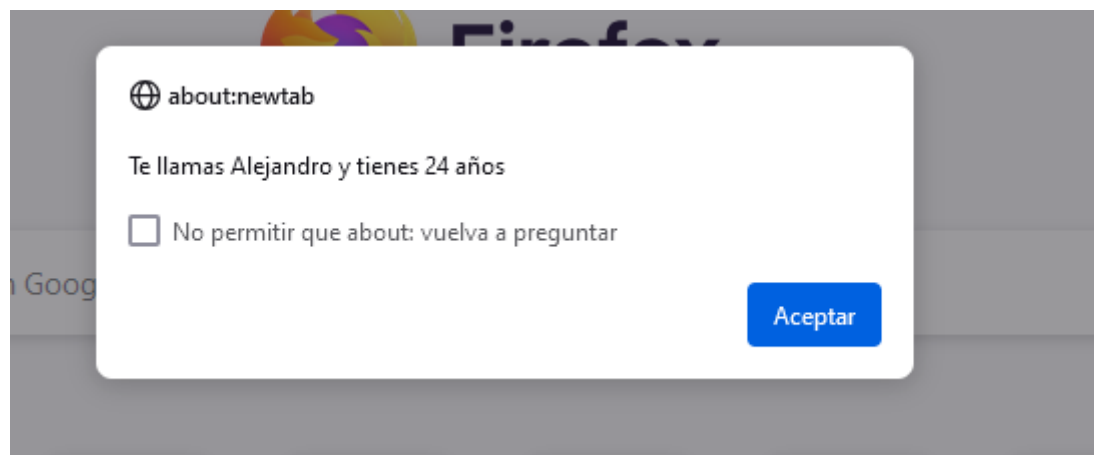
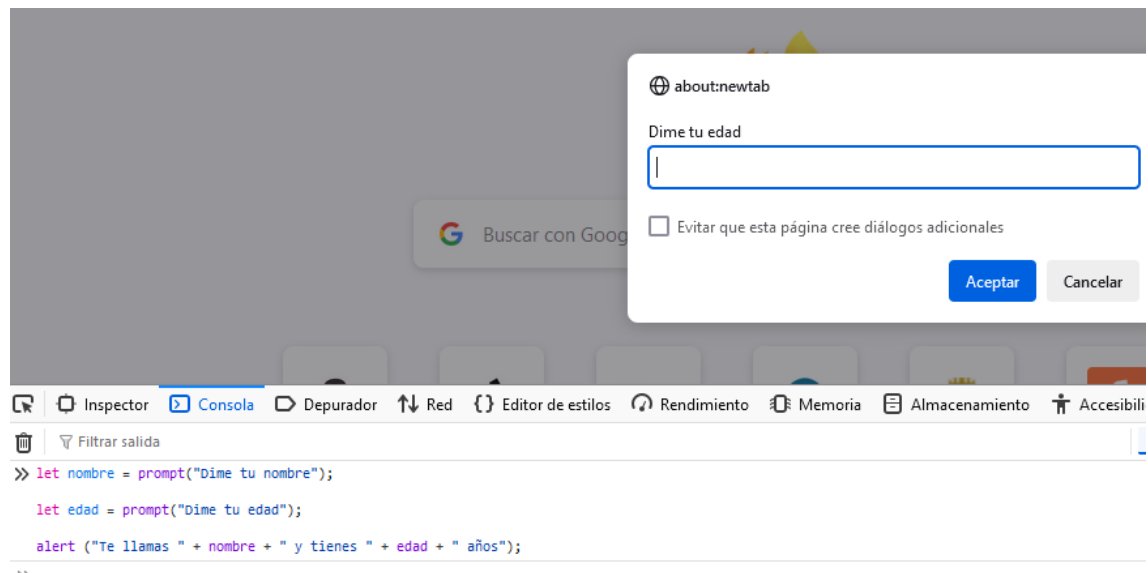
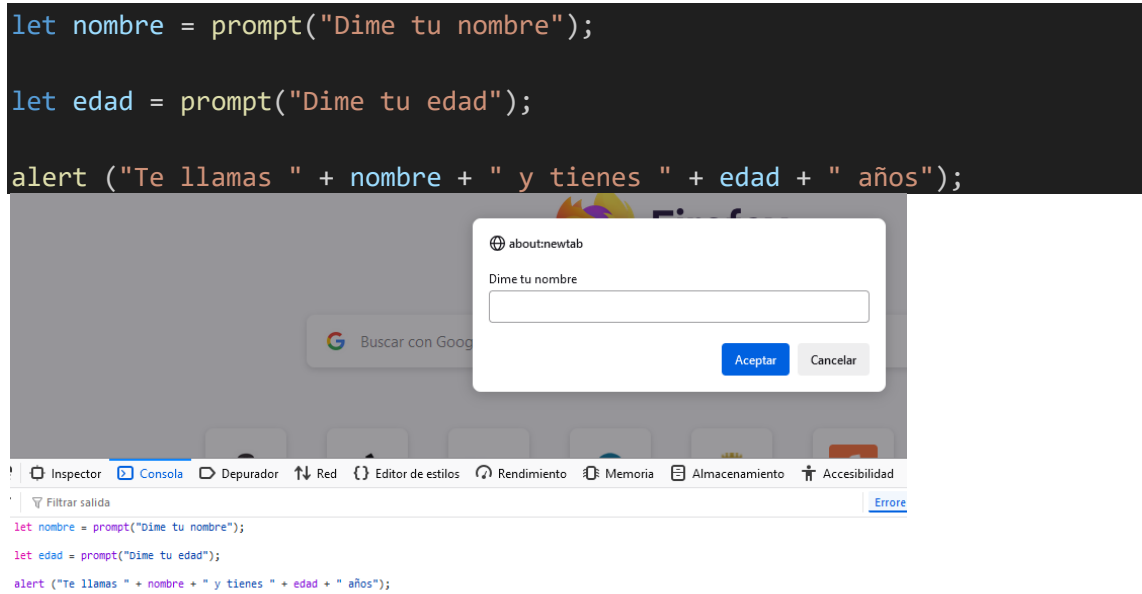


1)



2)

1)

```

>> 10 + 20
< 30
Suma de números

>> "10" + 20
< "1020"
Suma la cadena "10" y le añade el número 20

>> 10 + "20"
< "1020"
Como la anterior solo que ahora 10 es de tipo número y 20 es de tipo cadena.

>> +"10" + 20
< 30
Convierte el 10 que es cadena a número

>> +"10" + "20"
< "1020"
Convierte el 10 que es cadena a número y le añade la cadena 20.

```

2)

```

>> typeof(+"10" + "20") |
< "string"
Concatena el primer 10 que se convierte a número con la cadena 20.

>> typeof(+"10" + 20)
< "number"
Suma de dos número ya que el + antes del "10" lo cambia de string a number.

>> typeof(10 + "20")
< "string"
String ya que está concatenado un tipo number con el "20" que es de tipo string.

```

3)

```

>> let edadJubilacion = 65;
< undefined
Está declarando la variable.

>> typeof(edadJubilacion)
< "number"
Ahora diciendo de que tipo es la variable.

>> edadJubilacion += 2
< 67
Sumó 2 al valor declarado en la variable.

>> let edad = 17;
< undefined
Declara una nueva variable.

>> edad < edadJubilacion
< true
Compara si la segunda variable es menor que la primera, al ser correcto devuelve un true.

```

4)

```
>> let jubilacion = "60";
```

```
← undefined
```

Declara la variable.

```
>> typeof(jubilacion);
```

```
← "string"
```

Da de tipo string debido a que el 60 va entre comillas.

```
>> jubilacion == edadJubilacion;
```

```
← false
```

Compara 2 tipos distintos, un string y un number.

```
>> jubilacion === edadJubilacion;
```

```
← false
```

Compara de manera más estricta, aquí debe coincidir también el tipo.

5)

```
>> --edadJubilacion;
```

```
← 66
```

Resta a la edadJubilacion un año.

```
>> edadJubilacion -= 6;
```

```
← 60
```

Ahora resta 6 años más.

```
>> jubilacion = 60;
```

```
← 60
```

Da el valor 60 a jubilación.

```
>> edadJubilacion = jubilacion++ + 4;
```

```
← 64
```

Está diciendo que la edadJubilacion es la jubilación sumándole 4.

6)

```
>> let x = 4;
```

```
← undefined
```

Define que x tiene el valor de 4.

```
>> x = 6;
```

```
← 6
```

Ahora x tiene el valor de 6.

```
>> const constante = 7;
```

```
← undefined
```

Define una constante con el valor de 7.

```
>> constante = 0;
```

```
❗ ▶ Uncaught TypeError: invalid assignment to const 'constante'  
    <anonymous> debugger eval code:1  
    [Saber más]
```

No puede ya

que constante es 7, no puede cambiar.

7)

```

>> let edad = 30;
← undefined
>> console.log("TIENES\n" + edad + "\nAÑOS");
TIENES
30
AÑOS
← undefined

```

Usa el \n para saltar línea.

8)

```

>> let edad = 24;
← undefined
>> // Muestra el doble de tu edad

    console.log(`El doble de tu edad es: ${edad * 2}`);
El doble de tu edad es: 48
← undefined

```

Las comillas

simples sirven para poder poner variables dentro de una cadena.

3)


```

▼ let nombre = prompt("Dime tu nombre");

let edad = Number(prompt("Dime tu edad")); //convertimos la cadena a numero

if (edad < 18) {
    let annosFaltan = 18 - edad;
    alert("Te llamas " + nombre + " y tienes " + edad + " años. Te faltan " + annosFaltan + " años para ser mayor de edad.");
} else {
    alert("Te llamas " + nombre + " y tienes " + edad + " años.");
}

```

 about:newtab

Te llamas Alejandro y tienes 14 años. Te faltan 4 años para ser mayor de edad.

☐ No permitir que about: vuelva a preguntar

Aceptar

4)


```

let num1 = Number(prompt("Dime un número"));
let num2 = Number(prompt("Dime otro número"));
let num3 = Number(prompt("Dime otro número"));
let num4 = Number(prompt("Dime el último número"));

let suma = num1 + num2 + num3 + num4;
let producto = num1 * num2 * num3 * num4;

alert ("La suma de " + num1 + " + " + num2 + " + " + num3 + " + " + num4 + " es " + suma);
alert ("El producto de " + num1 + " * " + num2 + " * " + num3 + " * " + num4 + " es " + producto);


```

 about:newtab

La suma de $1 + 2 + 3 + 4$ es 10

☐ No permitir que about: vuelva a preguntar

Aceptar

 about:newtab

El producto de $1 * 2 * 3 * 4$ es 24

☐ No permitir que about: vuelva a preguntar

Aceptar