

3.5.0

Options

.....

37 specs, 0 failures

finished in 0.012s

## Pair Programming

## Declarando variables

- 1.- variable tipo let de nombre 'variablesSinvalor' está declarada sin valor
  - Es de tipo let
  - no tiene valor asignado
- 2.- 2 variables tipo let de nombres 'booleano1' y 'booleano2' está declarada y es de tipo booleano
  - booleano1 es de tipo let
  - booleano1 es booleano
  - booleano2 es de tipo let
  - booleano2 es booleano
- 3.- variable tipo const de nombre 'PI' está declarada con valor de 3,14
  - Es de tipo const
  - tiene valor 3.14
- 4.- variable tipo const de nombre 'TAU' está declarada con valor de 2 veces PI
  - Es de tipo const
  - tiene valor 2PI

## Booleanos

- 5.- variable 'booleanoAnd' cuyo valor sea la comparación booleana 'booleano1 and booleano2'
- 6.- variable 'booleanoNot' cuyo valor sea la comparación booleana 'no booleano1'
- 7.- variable 'booleanoMix0' cuyo valor sea la comparación booleana '(booleano1 or booleano2) and (booleano1 or (not booleano1 and not booleano2))'

## Operadores

- 8.- crea la variable 'incrementarDesp' con valor 2 y asigna su valor con postincremento a 'resultadoDesp'
  - incrementarDesp vale 2
  - resultadoDesp tienen el valor correcto
- 9.- crea la variable 'incrementarAntes' con valor 2 y asigna su valor con preincremento a 'resultadoAntes'
  - incrementarAntes vale 2
  - resultadoAntes tienen el valor correcto

## Bucles

- 10.- crea la variable 'contarHasta10\_2' con valor 0 e incrementa su valor con un bucle for hasta que se verifique contarHasta10\_2 === 10
- 11.- Crear las variables postI y postJ con valor 0 a continuación cree un bucle que itere 11 veces. En cada iteración se deberá sumar al valor de postI el valor de postJ++
- 12.- crea la variable 'sumaPares' con valor 0 a continuación crea un bucle que itere 10 veces (i < 10), si la iteración es par se deberá sumar a sumaPares el número de la iteración actual (i)

## proyecto

## Declarando variables

- 13.- variable tipo let de nombre 'variableValorNumerico' está declarada con valor numérico
  - Es de tipo let
  - tiene valor numérico
- 14.- variable tipo const de nombre 'MiNombre' está declarada con valor de tu nombre
  - Es de tipo const
  - tiene valor cadena de texto
- 15.- variable tipo const de nombre 'MiNumeroFav' está declarada con valor numérico
  - Es de tipo const
  - tiene valor numérico

## Booleanos

- 16.- variable 'booleanoOr' cuyo valor sea la comparación booleana 'booleano1 or booleano2'
- 17.- variable 'booleanoMix1' cuyo valor sea la comparación booleana '(booleano1 and TAU/2 sea igual a PI) or (variableValorNumerico mayor o igual que MiNumeroFav)'
- 18.- variable 'seisNoEsNueve' cuyo valor sea la comparación booleana '6 no es igual que 9'
- 19.- variable 'booleanoMix2' cuyo valor sea la comparación booleana 'variableValorNumerico positivo o menor que -(MiNumeroFav \* TAU)'

#### Operadores

- 20.- variable 'valorSuma' cuyo valor sea la suma de MiNumeroFav y variableValorNumerico
- 21.- variable 'valorResta' cuyo valor sea la resta de MiNumeroFav menos variableValorNumerico
- 22.- variable 'valorMultiplicacion' cuyo valor sea la multiplicación de MiNumeroFav por variableValorNumerico
- 23.- variable 'valorDivision' cuyo valor sea la division de MiNumeroFav entre 3

#### Bucles

- 24.- crea la variable 'contarHasta10' con valor 0 e incrementa su valor con un bucle while hasta que se verifique contarHasta10 === 10
- 25.- Crear las variables preI y preJ con valor 0 a continuación cree un bucle que itere 11 veces. En cada iteración se deberá sumar al valor de preI el valor de ++preJ
- 26.- crea la variable 'sumaImpares' con valor 0 a continuación crea un bucle que itere 10 veces (i < 10), si la iteración es impar se deberá asumir a sumaPares el número de la iteración actual (i)