**Getting Started**

All your work should be done in the Your Solution section.

More specifically, your task is to implement the function **werewolfCheck**. The function declaration is already provided in the Your Solution section.

Due to the testing environment setup, **you should not change the name of the provided function declaration werewolfCheck(name){ }**.

To begin working on the solution, you should start writing your code inside the function.

**Hint**: You should always first go over and read the *Task Instructions* carefully to ensure that you fully understand the task requirements.

**Task Instructions**

Werewolves are hiding among us! However, they are terrible at naming themselves. All their names start with "were", like for example "werebob", "wereandy" and "weresarah".

Implement a function named **werewolfCheck** that receives an argument name, which is a string and checks if it begins with "were".

If the provided ***name*** begins with were the function should **return** a string "it is a werewolf", otherwise it should return a string "just a regular person".

**Important:** You must use the keyword return to ***return the value from the function***. Tests can only check the output value that your function returns. Any console.log output will be ignored.

⠀

**Example - Input/Expected Output:**

**input**: "werebrian"  
**output**: "it is a werewolf"

**input**: "weretasha"  
**output**: "it is a werewolf"

**input**: "anna"  
**output**: "just a regular person"

function werewolfCheck (name) {

let name = 'were';

if (name === 'were'){

return "it is a werewolf"

} else (name !== "were")

{ return "Just a regular person"

}

}

# 1.3 - JS Challenge | Days in a Month

## Context

This challenge tests your understanding of **primitive data types, conditionals and function basics**.

## Getting Started

All your work should be done in the Your Solution section.  
More specifically, your task is to implement the function **daysInMonth**. The function declaration is already provided in the Your Solution section.

Due to the testing environment setup, **you should not change the name of the provided function declaration daysInMonth (month){ }**.

To begin working you just need to start writing your code inside the switch block where indicated by // Your code here.

**Hint**: You should always first go over and read the Task Instructions carefully to ensure that you fully understand the task requirements.

## Task Instructions

Implement the logic of the function named **daysInMonth**. The function receives one argument month, which is an integer (whole number) in the range from 1 to 12.

It should return a number of days depending on the month of the year, as per the given table:

+-----------------+------------+

| month | days |

+-----------------+------------+

| 1,3,5,7,8,10,12 | 31 |

+-----------------+------------+

| 4,6,9,11 | 30 |

+-----------------+------------+

| 2 | 28 |

+-----------------+------------+

**Important:** You must use the keyword return to **return the value from the function**. Tests can only check the output value that your function returns. Any console.log output will be ignored.

#### Example - Input/Expected Output:

**input**: 1  
**output**: 31

**input**: 6  
**output**: 30

**input**: 2  
**output**: 28

Disclaimer: Ignore the leap years in this example!

## Running Your Code

To run the code and to check your progress using the **RUN TESTS** button.  
This will run your code against the tests and output the results in the Test Results section.

The Test Results section contains the list of all the tests that your code Passed or Failed. These test messages are requirements that your code should fulfill. They include a description of what you have to do.

### Challenge Submission

**When ready you can submit the current challenge solution by clicking the *SUBMIT SOLUTION* button**.

### Final Submission

To do a final submission of the entire assessment, go to the last challenge in the assessment and click on the **REVIEW ASSESSMENT** button in the upper right corner.

There you will be prompted to check your code once again and do a final submission.

**Note:** You can't do any re-submission after this step is done.

Good luck!

Your Ironhack team

function daysInMonth(month){

let month = '1,2,3,4,5,6,7,8,9,10,11,12';

let days = '28,30,31';

switch (month){

case 1, 3, 5, 7, 8, 10, 12:

return '31';

case 4,6,9,11:

return 30;

case 2:

return '28'

}

}

## Context

This challenge tests your understanding of how to use **strings, loops, and conditionals**.

## Getting Started

All your work should be done in the Your Solution section.  
More specifically, your task is to implement the function **countSmilies**.  
The function declaration is already provided in the Your Solution section.

Due to the testing environment setup, **you should not change the name of the provided function declaration countSmilies (message) { }**.

To begin working on the solution, you should start writing your code inside the function.

**Hint**: You should always first go over and read the Task Instructions carefully to ensure that you fully understand the requirements of the task.

## Task Instructions

A friend of a friend is addicted to messaging with smiley faces! 😃

Implement the function **countSmilies** that has one parameter - the message.  
The function should return the count of how many times a smiley face :) appears in the message.

**Hint:** Remember that :) is two characters, so you might need to check both in separate conditions :) .

**Important:** You must use the keyword return to **return the value from the function**. Tests can only check the output value that your function returns. Any console.log output will be ignored.

#### Example - Input/Expected Output:

**input**: "Hello :), how are you doing :) :D"  
**output**: 2

**input**: "To be :D or not to be :D =)"  
**output**: 0

**input**: "It is raining :(, but I don't care :) :) :). And you ? :-)"  
**output**: 3

## Running Your Code

To run the code and to check your progress using the **RUN TESTS** button.  
This will run your code against the tests and output the results in the Test Results section.

The Test Results section contains the list of all the tests that your code Passed or Failed. These test messages are requirements that your code should fulfill. They include a description of what you have to do.

### Challenge Submission

**When ready you can submit the current challenge solution by clicking the *SUBMIT SOLUTION* button**.

# 2.3 - JS Challenge | Pick Odd Numbers

## Context

This challenge tests your understanding of working with **loops and arrays**.

## Getting Started

All your work should be done in the Your Solution section.  
More specifically, your task is to implement the function **pickOdd**. The function declaration is already provided in the Your Solution section.

Due to the testing environment setup, **you should not change the name of the provided function declaration pickOdd (arr) { }**.

To begin working on the solution, you should start writing your code inside the function where indicated by // Your code here.

**Hint**: You should always first go over and read the Task Instructions carefully to ensure that you fully understand the task requirements.

## Task Instructions

Implement the function **pickOdd** that has one parameter arr, which is an array of numbers.

The function should loop over the array arr and check each number in the array if it is an even or odd number.  
If the number is odd (1,3,5, etc...), it should be added to the array named oddNumbers that is already provided in the function.

Array oddNumbers is already included in the function, as well as the return oddNumbers; statement. **Do not change this.**  
This is the reason some of the tests are already Passing.

Your job is to create a for loop adding odd numbers to the oddNumbers array.

**Important:** You must use the keyword return to **return the value from the function**. Tests can only check the output value that your function returns. Any console.log output will be ignored.

#### Example - Input/Expected Output:

⠀  
**input**: [1, 2, 3, 4]  
**output**: [1, 3]

**input**: [1, 1, 3, 21]  
**output**: [1, 1, 3, 21]

**input**: [2, 10, 2, 6]  
**output**: []

## Running Your Code

To run the code and to check your progress using the **RUN TESTS** button.  
This will run your code against the tests and output the results in the Test Results section.

The Test Results section contains the list of all the tests that your code Passed or Failed. These test messages are requirements that your code should fulfill. They include a description of what you have to do.

### Challenge Submission

**When ready, you can submit the current challenge solution by clicking the *SUBMIT SOLUTION* button**.

### Final Submission

To make a final assessment submission, go to the last challenge in the assessment and click on the **REVIEW ASSESSMENT** button in the upper right corner.

There you will be prompted to check your code once again and do a final submission.

**Note:** You can't do any re-submission after this step is done.

Good luck!

Your Ironhack team

function pickOdd (arr) {

const oddNumbers = []; // do not remove or change this line

var i = [44,6,7,2,1,39]

if ( i % 2) == 1) {

}

return oddNumbers; // do not remove or change this line

}