## If Else statements

What is an If Else statement and how do we use it?

If Else statements are a part of the programming fundamentals. Basically, they tell a statement, that “if…”, “or else…” but you can use it much more advanced than that. In our game, we want to use it to:

1. Check which player guesses correctly, give them a point and reset the turn.
2. Declare a winner and reset the game, so that we can play again.

But how do we do that?

Let’s start with the first one, checking which player guesses correctly and gains a point. So why do we want to use an if else statement for this? Because it makes sense. Why make something complicated, when we can just say “If player1 guesses correctly, give them a point. Else, don’t”?

To make this, we first make the statement.

*if* ($("player1High").checked) {

}

This statement says: If player1High is checked, do something. This is the base of our if statement, because we want to be able to check if the player clicked.

We want the function to be able to check if we guessed right, which we will do so:

*if* ($("player1High").checked) {

*if* (arr[arr.length - 1] > arr[arr.length - 2]) {

console.log("correct");

$("player1Points").innerHTML++;

} *else* {

console.log("false");

}

}

*if* ($("player1Low").checked) {

*if* (arr[arr.length - 1] < arr[arr.length - 2]) {

console.log("correct");

$("player1Points").innerHTML++;

} *else* {

console.log("false");

}

}

In this snippet, we make an if statement within an if statement. We do this, because we want to make sure that it knows if you guessed correctly or not. We do this by making a correct or false answer which says “If(guess) is correct, give a point to player1Points innerHtml and if it’s not correct, do not give a point.

This is now our first if-else statement, which makes sure that we are able to get points if we guess correctly.

For our game, we want to be able to use these commands, when we press the roll button, so that the gameplay is smooth, and so that when we eventually do get a point, it resets so that we can use it again. We do this by wrapping it in a function, so that we can easily access it later when we need it. Let’s call it player1Turn. After our if else statement, we need to reset the players turn, which we will do by setting id”).checked = false;

let player1Turn = () => {

*if* ($("player1High").checked) {

*if* (arr[arr.length - 1] > arr[arr.length - 2]) {

console.log("correct");

$("player1Points").innerHTML++;

} *else* {

console.log("false");

}

}

*if* ($("player1Low").checked) {

*if* (arr[arr.length - 1] < arr[arr.length - 2]) {

console.log("correct");

$("player1Points").innerHTML++;

} *else* {

console.log("false");

}

}

$("player1Low").checked = false;

$("player1High").checked = false;

}

This is now our if else statement which we put inside a function. This checks if the player pressed the higher or lower button and decides which one is the correct one and allocates points depending on the result. We can copy this code snippet and say player2x instead of the current ids, so that we target the player 2 in our HTML.

Now we have made a function, which has if else statements in it which will determine which player determines correctly, give them a point and reset the turn. Now we need to declare a winner and reset the game. This we will do simply by saying “If player1 has 3 points, player 1 wins, or else if player 2 has 3 points, player 2 wins.

let winner = () => {

*if*($("player1Points").innerHTML == 3){

alert("Player 1 WINS!");

newGame();

}

*else* *if*($("player2Points").innerHTML == 3){

alert("Player 2 WINS!");

newGame();

}

}

In the snippet above, we use an alert and call the “newGame” function. The alert is used to announce who the player is, but you will get a bit more information about that later.

The “newGame” function is a function we made earlier, which set the points to 0 and resets the roll mechanism in the game.

## Alerts

An alert is a pop-up, that appears when executed in your script. This could be used in many ways but is often used to make sure information gets properly through to the user. In our game, it would be nice to know who wins, right? Making an alert is simple, the only code you need to write is:

function useAlert(){

alert("This is our alert message !");

}

Using this function with a for example a button, would trigger an alert which states “This is our alert message!”.

It looks quite simple, and we used it in our lesson about “If Else Statements” to declare who is the winner, here is the code snippet:

let winner = () => {

*if*($("player1Points").innerHTML == 3){

alert("Player 1 WINS!");

newGame();

}

*else* *if*($("player2Points").innerHTML == 3){

alert("Player 2 WINS!");

newGame();

}

}

In this If else statement we simply put an alert method inside the statement, to declare who wins based on who reaches 3 points first.

## A simple trick can make all the difference

Throughout our code, we have been using $ like this:

$("roll");

This is a smart way of writing code, by decreasing the amount of code you need to write, every time you need to fetch an id from your HTML. We make the $ into a function:

let $ = (x) => {

*return* document.getElementById(x);

}

This is our $ function. In this function we make it return the getElementById method, one of the most used methods (which we use to fetch ids from our html), to cutting down the time it takes to write it.

*Simple yet effective.*

For an example on how we use this to make our game, you can look at this code snippet from a previous lesson, where we check which player has guessed correctly and receives a point:

let player1Turn = () => {

*if* ($("player1High").checked) {

*if* (arr[arr.length - 1] > arr[arr.length - 2]) {

console.log("correct");

$("player1Points").innerHTML++;

} *else* {

console.log("false");

}

}

*if* ($("player1Low").checked) {

*if* (arr[arr.length - 1] < arr[arr.length - 2]) {

console.log("correct");

$("player1Points").innerHTML++;

} *else* {

console.log("false");

}

}

$("player1Low").checked = false;

$("player1High").checked = false;

}