Code Challenges: JavaScript Fundamentals (Variables, Data Types, Conditionals, Functions

Solution: greetWorld

```
const greetWorld = function(){
  return "Hello, World!"
}
```

Solution: canIVote

```
const canIVote = function(age){
  if (age >= 18){
    return true;
  }else{
    return false;
  }
}
```

Solution: agreeOrDisagree()

```
const agreeOrDisagree = function(firstString, secondString){
  if(firstString === secondString){
    return "You agree!";
  }else{
    return "You disagree!";
  }
}
```

Solution: lifePhase

```
const lifePhase = function(age){
if(age < 0 || age > 140){
  return "This is not a valid age"
}else if(age <= 3){ // It’s the same: age < 4
  return "baby"
}else if(age <= 12 ){ // age < 13
  return "child"
}else if(age <= 19){  // age < 20
```

```
  return "teen"
}else if(age <= 64){ // age < 65
  return "adult"
}else{
  return "senior citizen"
}
};
```

Arrow Function example:

```
const finalGrade = (midterm, final, homework) => {
```

Solution: finalGrade

```
const finalGrade = function(homework, assignment, test){
   if ((homework < 0 || homework > 100) || (assignment < 0 || assignment >
 100) || (test < 0 || test > 100)){
    return "You have entered an invalid grade."
}
  const average = (homework + assignment + test) / 3;
  if (average < 60){
    return "F"
  }else if (average < 70){
    return "D"
  }else if (average < 80){
    return "C"
  }else if (average < 90){
    return "B"
  }else{
    return "A"
  }
};
console.log(finalGrade(99, 92, 95))
```

***Solution**: reportingForDuty*

```
const reportingForDuty = function(rank, lastName){
  return `${rank} ${lastName} reporting for duty!`
}
console.log(reportingForDuty("Private", "Fido"))
```

***Solution**: rollTheDice*

```
const rollTheDice = () => {
  let die1 = Math.floor(Math.random() * 6 + 1);
  let die2 = Math.floor(Math.random() * 6 + 1);
  return die1 + die2;
};
console.log(rollTheDice("die1", "die2"))
```

***Solution** calculateWeight:*

There is no point to using break keyword as the return keyword stops the execution and anything performed after the return keyword won't be seen. In other words, nothing after return is reachable.

```
function calculateWeight(earthWeight, planet){
  switch(planet){
    case "Mercury":
    return earthWeight * 0.378;
    case "Venus":
    return earthWeight * 0.907;
    case "Mars":
    return earthWeight * 0.377;
    case "Jupiter":
    return earthWeight * 2.36;
    case "Saturn":
    return earthWeight * 0.916;
    default:
    return "Invalid Planet Entry. Try: Mercury, Venus, Mars, Jupiter, or Saturn."
  }
}
```

***Solution**: truthyOrFalsy*

```
const truthyOrFalsy = function(value){
  if(value){
    return true;
  }else{
    return false
  }
}
console.log(truthyOrFalsy(5));
console.log(truthyOrFalsy(0));
// falsy: null, undefined, "", Nan, 0, false,
// Using Ternary:
// const truthyOrFalsy = value => value ? true : false
```

Solution: numImaginaryFriends

```
const numImaginaryFriends = function(totalFriends){
  return Math.ceil(totalFriends * .25);
}
//console.log(numImaginaryFriends(10));
```

Solution: sillySentence

```
const sillySentence = function(oneString, secString, thirString){
  return `I am so ${oneString} because I ${secString} coding! Time to write some more awesome ${thirString}!`
}

//console.log(sillySentence('excited', 'love', 'functions'));
```

Solution: howOld

```
const howOld = (age, year) => {
    let dateToday = new Date();
    let currentYear = dateToday.getFullYear();
    // To get the today's year use the above, instead of typing let currentYear = 2022
    const yearDifference = year - currentYear
    const newAge = age + yearDifference

    if (newAge > age) {
        return `You will be ${newAge} in the year ${year}`
    } else if (newAge < 0) {
        return `The year ${year} was ${-
newAge} years before you were born`
    } else {
        return `You were ${newAge} in the year ${year}`
    }
}
console.log(howOld(35, 2036))
```

Solution: whatRelation

```
const whatRelation = percentSharedDNA => {
    if(percentSharedDNA === 100){
      return 'You are likely identical twins.'
    }else if(percentSharedDNA > 34){
      return 'You are likely parent and child or full siblings.'
    }else if(percentSharedDNA > 13){
      return 'You are likely grandparent and grandchild, aunt/uncle and ni
ece/nephew, or half siblings.'
    }else if(percentSharedDNA > 5){
      return 'You are likely 1st cousins.'
    }else if(percentSharedDNA > 2){
      return 'You are likely 2nd cousins.'
    }else if(percentSharedDNA > 0){
      return 'You are likely 3rd cousins'
    }else{
      return 'You are likely not related.'
      }
    }

console.log(whatRelation(34))
// Should print 'You are likely grandparent and grandchild, aunt/uncle and
 niece/nephew, or half siblings.'

console.log(whatRelation(3))
// Should print 'You are likely 2nd cousins.'
```

Solution: tipCalculator

```
const tipCalculator = function(quality, total){

switch(quality){
    case "bad":
    return total * 0.05;
    case "ok":
    return total * 0.15;
    case "good":
    return total * 0.2;
    case "excellent":
    return total * 0.3;
    default:
    return total * 0.18;
}
}
console.log(tipCalculator('good', 100)) //should return 20
```

Change it from switch to if/else statement:

```
const tipCalculator = function(quality, total){
  if(quality === "bad"){
    return total * 0.05
  } else if(quality === "ok"){
    return total * 0.15
  } else if(quality === "good"){
    return total * 0.20
  } else if(quality === "excellent"){
    return total * 0.30
  }else{
    return total * 0.18
  }
}
```

Solution: toEmoticon

```
const toEmoticon = function(emotions){
  switch(emotions){
  case 'shrug':
  return '|_{"}_|'
  case 'smiley face':
  return ':)'
  case 'frowny face':
  return ':('
  case 'winky face':
  return ';)'
  case 'heart':
  return '<3'
  default:
  return "|_(* ~ *)_|"
}
  }
```

Solution: Code Challenge 1 - colorMessage

```
const colorMessage = function(favoriteColor, shirtColor){
  if(favoriteColor === shirtColor){
    return "The shirt is your favorite color!"
  }else{
    return "That is a nice color."
  }
}
console.log(colorMessage("blue", "blue"));
```

Solution: Code Challenge 2 - isEven

```
const isEven = function(num){
  if(num % 2 == 0){
    return true
  }else{
    return false
  }
}
console.log(isEven(5));
```

Solution: Code challenge 3 - numberDigits

My working solution with return keyword:

```
const numberDigits = function(num){
  if(num >=0 && num < 10){
    return `One digit: ${num}`
  }else if(num >=10 && num < 100){
    return `Two digits: ${num}`
  }else{
    return `The number is: ${num}`
  }
}
console.log(numberDigits(-520));
```

The other Solution below without return keyword, instead create variable numString and set it equal to empty string.

Challenge was: If the variable x is between 0 and 9, return the string 'One digit: N', where N is the value of x. If the variable x is between 10 and 99, return the string 'Two digits: N', where N is the value of x. Any other value of x, including negative numbers, return the string 'The number is: N', where N is the value of x.

```
const numberDigits = x => {
  let numString = '';
  if (x >= 0 && x <= 9){
    numString = 'One digit: ' + x;
  }else if (x >=10 && x <= 99){
    numString = 'Two digits: ' + x;
  }else {
    numString = 'The number is: ' + x;
  }

  return numString;
}
```