Variables and Expressions

Overview

```
/*
  - Assigning values to variables
  - Expressions
  - Shortcut syntax
*/
```

Variable assignment

```
1  // values can be stored in a variable
2  // use the assignment operator =
3
4  let myNum = 5;
5  console.log(myNum);
6
7
8
```

overjoyed

Variable assignment

```
1  /* the let keyword stores the assigned value
2     in a variable. the variable can be
3     reassigned */
4
5  let mood = 'happy';
6  mood = 'overjoyed';
7
8  console.log(mood);
```

SyntaxError: Assignment to constant variable

Variable assignment

```
1  /* the const keyword stores the assigned
2     value in a constant variable that cannot
3     be reassigned */
4
5  const favoriteBootcamp = 'TEJ';
6
7  // this line will throw an error
8  favoriteBootcamp = 'somewhere else';
```

Expressions

```
15
false
happy together
```

```
1  /* an expression is 'any valid unit of code
2     that resolves to a value' (MDN) */
3
4  console.log(15);
5
6  console.log(9 === 10);
7
8  console.log('happy' + ' ' + 'together');
```

Expressions

```
1  /* since expressions result in a value,
2     expressions can be assigned to variables*/
3
4  let sum = 10 + 5;
5  console.log(sum);
6
7
8
```

Evaluating expressions

```
1  /* to write code that works, we have to
2     anticipate how JS will evaluate our
3     expressions */
4
5  // remember 'order of operations' from math?
6  let notSure = 10 + 5 * 10;
7
8  console.log(notSure);
```

Evaluating expressions

```
1  /* JS has a similar concept to order of
2     operations called operator precedence */
3
4  /* mathematic operators are applied in the
5     same order in mathematics and JS */
6  let notSure = 10 * 10 / (5 * 5);
7
8  console.log(notSure);
```

Evaluating expressions

```
1  /* JS has a similar concept to order of
2     operations called operator precedence */
3
4  /* mathematic operators are applied in the
5     same order in mathematics and JS */
6  let notSure = 10 * 10 / 5 * 5;
7
8  console.log(notSure);
```

true

Evaluating expressions

```
1  /* as in math, any JS expression wrapped in
2     parentheses will be evaluated first */
3
4  let notSure = false === (10 !== 10);
5
6  console.log(notSure);
7
8
```

false

Evaluating expressions

```
1  /* there are well-defined rules describing
2     the operator precedence for every operator
3     in JavaScript! just search MDN operator
4     precedence when you need it; no need to
5     memorize */
6
7  let notSure = typeof 10 === 10;
8  console.log(notSure);
```

```
sum: 15
```

Shortcut syntax

```
1  // we can assign new values to a variable
2  let sum = 0;
3  sum = sum + 5;
4  sum = sum + 10;
5
6  console.log('sum:', sum);
7
8
```

sum: 15

Shortcut syntax

```
1  // the += operator sums and reassigns
2  // -=, *=, and /= all work the same way
3  let sum = 0;
4  sum +=  5;
5  sum += 10;
6
7  console.log('sum:', sum);
8
```

Shortcut syntax

```
1  /* the ++ operator increments by 1 and reassigns*/
2  let myNum = 0;
3  myNum++;
4
5  console.log(TEJ is #', myNum);
6
7
8
```

Shortcut syntax

```
1  /* the -- operator decrements by 1 and
2     reassigns */
3  let myNum = 3;
4  myNum--;
5  myNum--;
6
7  console.log(TEJ is #', myNum);
8
```

Running the test

```
>cd cd 03-expressions-variables-and-tdd
>npm install
>npm run test

- What is package.json
- What is node_modules
- What does npm install do?
- What does npm run script do?


```

Recap

```
/*
  - Assigning values to variables
  - Expressions
  - Shortcut syntax
*/
```