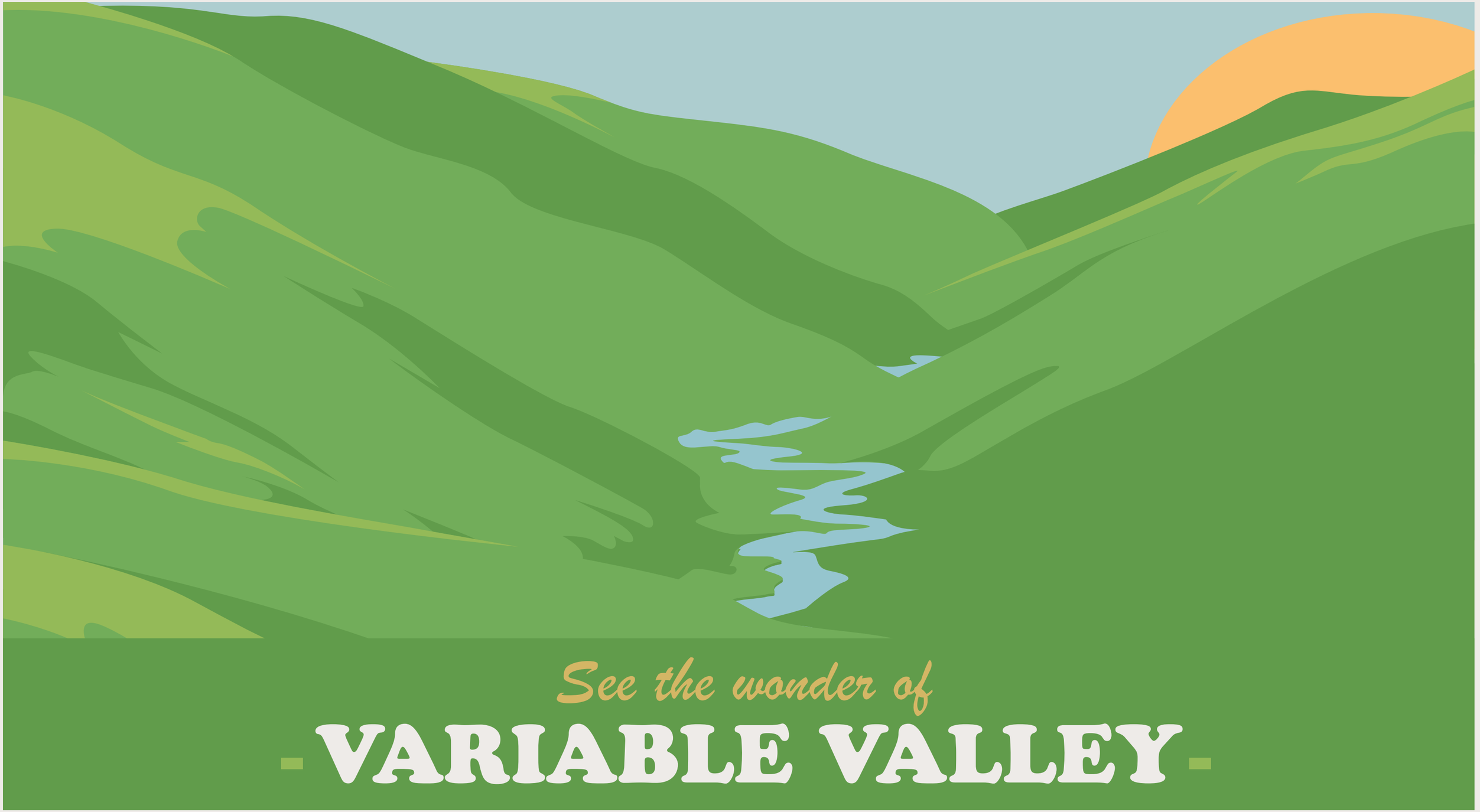
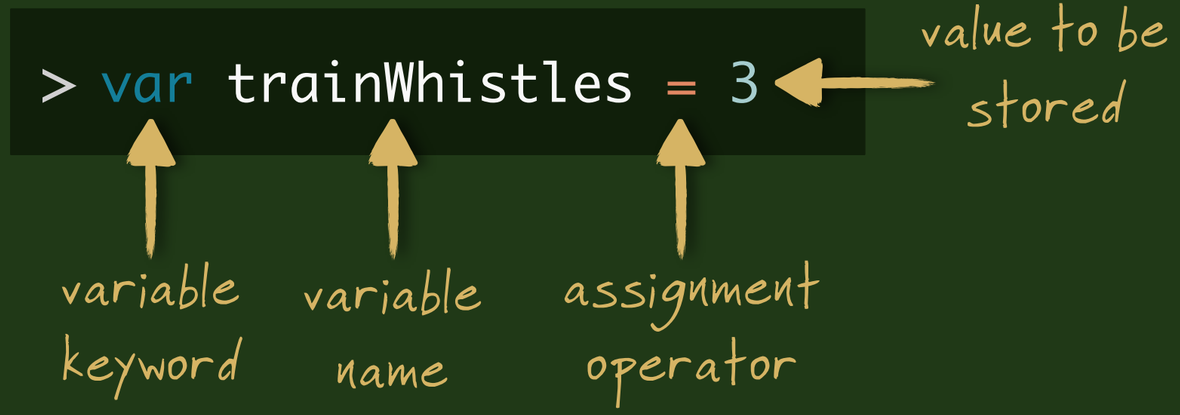

See the wonder of
VARIABLE VALLEY

LEVEL 2

VARIABLE VALLEY

STORING OUR VALUES

JavaScript uses variables to store and manage data



NAMING VARIABLES

Rules and regulations

```
var no spaces
```

no spaces in the name

```
var 3blindmice
```

no digits in front

```
var scored_is_fine
```

underscores are okay, but often irritating

```
var get$
```

dollar signs are also cool...but don't be that person

```
var $_$
```

slightly stupid, but technically legal

```
var goodName
```

begin with lowercase, later words capitalized, "camel case"

```
var mortalKombat2
```

FATALITY!!

CHANGING VARIABLE CONTENTS

Want to change a Variable's value? It's your lucky day.

```
>var trainWhistles = 3
```

```
> trainWhistles
```

→ 3

```
>trainWhistles = 9
```

no 'var' keyword this time, because JavaScript already "knows" about the variable

```
> trainWhistles
```

→ 9

```
>trainWhistles = trainWhistles + 3
```

uses current value to calculate new value

```
> trainWhistles
```

→ 12

CHANGING VARIABLE CONTENTS

Want to change a Variable's value? It's your lucky day.

```
> trainWhistles = trainWhistles + 3
```

same operation, different syntax

```
> trainWhistles += 3
```

```
> trainWhistles
```

→ 12

```
> trainWhistles
```

→ 15

CHANGING VARIABLE CONTENTS

Want to change a Variable's value? It's your lucky day.

```
> trainWhistles += 3
```

```
> trainWhistles
```

→ 15

```
> trainWhistles = trainWhistles * 2
```

same operation, different syntax

```
> trainWhistles
```

→ 30

```
> trainWhistles *= 2
```

```
> trainWhistles
```

→ 60

That's, like, a lot of whistles.

USING VARIABLES

Variable names also act as substitutes for the data they point to

```
> trainWhistles = 3
```

```
> "All of our trains have " + trainWhistles + " whistles!"
```

→ `"All of our trains have 3 whistles!"`

```
> "But the Pollack 9000 has " + (trainWhistles * 3) + "!"
```

→ `"But the Pollack 9000 has 9!"`

USING VARIABLES

Variable names also act as substitutes for the data they point to

```
> trainWhistles = 3
```

```
> "But the Pollack 9000 has " + (trainWhistles * 3) + "!"
```

```
> var pollack9000 = trainWhistles * 3
```

```
> pollack9000
```

→ 9

USING VARIABLES

Variable names also act as substitutes for the data they point to

```
> trainWhistles = 3
```

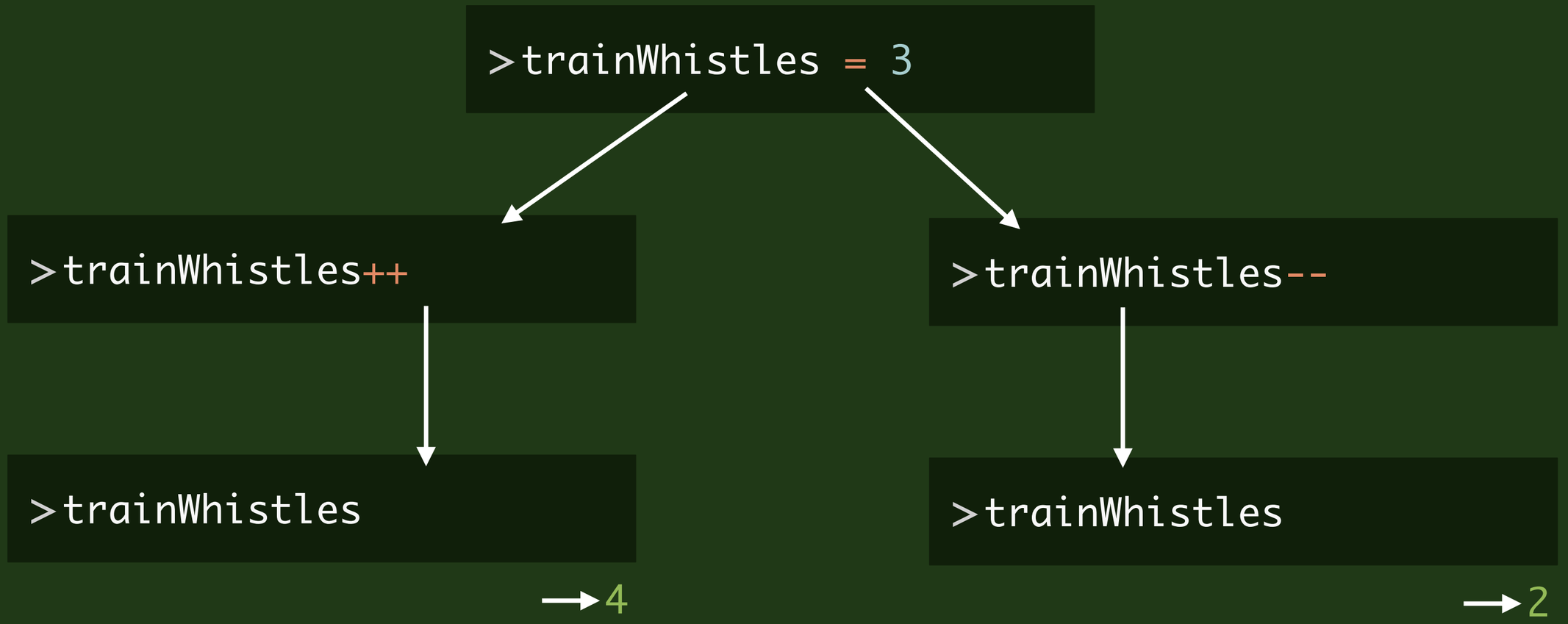
```
> var pollack9000 = trainWhistles * 3
```

```
> "But the Pollack 9000 has " + pollack9000 + "!"
```

→ `"But the Pollack 9000 has 9!"`

INCREMENTING AND DECREMENTING

A simple syntax for increasing or decreasing a variable's value by 1



VARIABLES STORE STRINGS, TOO!

JavaScript can store anything in variables.

```
> var welcome = "Welcome to the JavaScript Express Line!"
```

```
> var safetyTip = "Look both ways before crossing the tracks."
```

```
> welcome + "\n" + safetyTip
```

→ "Welcome to the JavaScript Express Line!
→ Look both ways before crossing the tracks."

USING VARIABLE NAMES WITH STRINGS

Variable names can also access the length property

```
> var longString = "I wouldn't want to retype this String every time."
```

```
> longString.length
→ 49
```

If a variable holds a String, we can access the length property directly from the variable name.

MORE COMPARISONS WITH VARIABLES

Comparing String lengths using the length property

```
> var longWordOne = "antidisestablishmentarianism"
```

```
> var longWordTwo = "supercalifragilisticexpialidocious"
```

Compares two numbers returned by the length properties

```
> longWordOne.length > longWordTwo.length
```

→ false

FINDING SPECIFIC CHARACTERS WITHIN STRINGS

Each position in a String has a numbered “index” starting from 0

```
> var sentence = "Antidisestablishmentarianism is fun to say!"
```

0 8 28 42

Think of index numbers as being a "distance from the starting character." Thus, the first character is "zero" away from itself.

Spaces are characters too!

There will always be one less index number than characters present!

```
> sentence.length
```

→ 43

Since the index starts at zero, but the length is counted by number of characters, the length value will always be one more than the last index.

FINDING SPECIFIC CHARACTERS WITHIN STRINGS

Each position in a String has a numbered “index” starting from 0

```
> var sentence = "Antidisestablishmentarianism is fun to say!"
```

```
> sentence.charAt(11)
```

→ `"b"`

```
> sentence.charAt(31)
```

→ `" "`

```
> sentence.charAt(42)
```

→ `"!"`

The charAt( ) method retrieves the character at a specific index.

VARIABLES HELP ORGANIZE DATA

Creating a versatile message out of flexible pieces

```
> var trainsOperational = 8
```

```
> var totalTrains = 12
```

```
> var operatingStatus = " trains are operational today."
```

```
> trainsOperational + " out of " + totalTrains + operatingStatus
```

→ `"8 out of 12 trains are operational today."`

VARIABLES HELP ORGANIZE DATA

Creating a versatile message out of flexible pieces

```
> var trainsOperational = 10
```

```
> var totalTrains = 12
```

```
> var operatingStatus = " trains are operational today."
```

```
> trainsOperational + " out of " + totalTrains + operatingStatus
```

→ `"10 out of 12 trains are operational today."`