

“THE LAST GREAT UNCHARTED FRONTIER”

Visit

THE CLIFFS OF VALUE



LEVEL 1

THE CLIFFS OF VALUE

THE PROMPT & SOME BASIC NUMBERS

- > The JavaScript Prompt, aka “the Console”
- What gets returned from the code

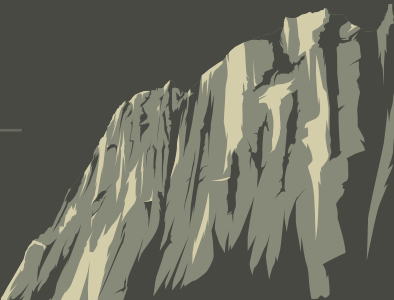
JavaScript automatically recognizes numbers

```
> 24
```

→ 24

```
> 3.14
```

→ 3.14



OPERATORS

Common Operators used in JavaScript Syntax:

addition

> 6 + 4

→ 10

subtraction

> 9 - 5

→ 4

multiplication

> 3 * 4

→ 12

division

> 12 / 4

→ 3

modulus

> 43 % 10

→ 3

Modulus returns the
remainder after division.

ORDER OF OPERATIONS: PEMDAS

Grouping Expressions in JavaScript

> (5 + 7) * 3

12 * 3

→ 36

> (3 * 4) + 3 - 12 / 2

12 + 3 - 12 / 2

12 + 3 - 6

→ 9

> (-5 * 6) - 7 * -2

-30 - 7 * -2

-30 - -14

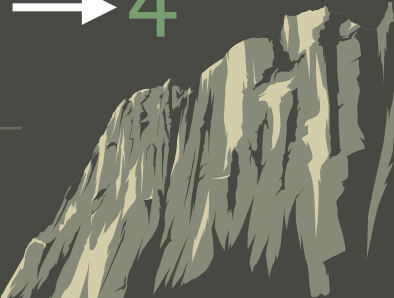
→ -16

> 4 + (8 % (3 + 1))

4 + (8 % 4)

4 + 0

→ 4



COMPARATORS

Common Number Comparators used in JavaScript Syntax:

greater than

`> 6 > 4`

→ true

"boolean" value



less than

`> 9 < 5`

→ false

greater or equal

`> 8 >= -2`

→ true

equals

`> 3 == 4`

→ false

two equal signs!



not equals

`> 12 != 4`

→ true

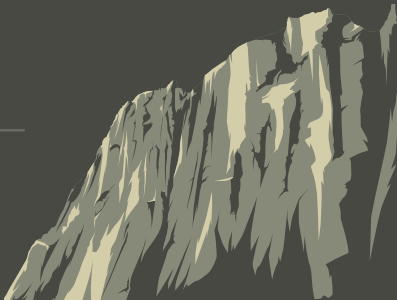
the "not" symbol



less or equal

`> 10 <= 10`

→ true



STRINGS

How JavaScript stores and processes flat text

```
> "Raindrops On Roses"
```

→ "Raindrops On Roses"

```
> "Whiskers On Kittens"
```

→ "Whiskers On Kittens"

Plus will glue Strings together

```
> "Raindrops On Roses" + " And " + "Whiskers On Kittens"
```

→ "Raindrops On Roses And Whiskers On Kittens"

Strings need quotes!

THESE ARE A FEW OF MY FAVORITE...STRINGS

Concatenation works with numbers and their expressions, too.

```
> "The meaning of life is" + 42
```

```
→ "The meaning of life is42"
```

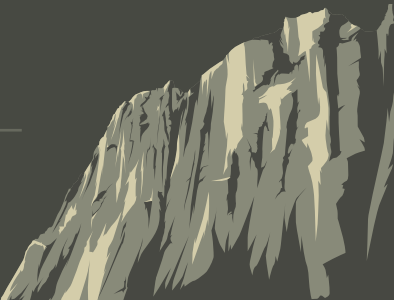


Uh oh...what happened?
Concatenation adds no
spaces, so we need to
add our own.

Notice the extra space!

```
> "The meaning of life is " + 42
```

```
→ "The meaning of life is 42"
```



THESE ARE A FEW OF MY FAVORITE...STRINGS

Concatenation works with numbers and their expressions, too.

```
> "Platform " + 9 + " and " + 3/4
```

```
→ "Platform 9 and 0.75"
```



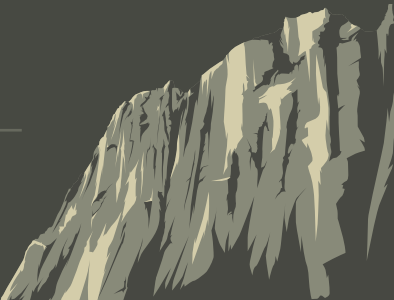
Expressions get evaluated!

```
> "Platform " + 9 + " and 3/4"
```

```
→ "Platform 9 and 3/4"
```



Make strings out of expressions that you want to see in their original format.



SPECIAL CHARACTERS INSIDE STRINGS

Some characters need backslash notation in JavaScript Strings

advances to the next "tab stop"

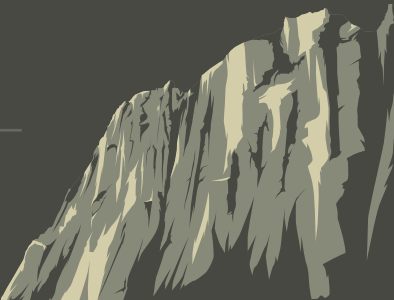
```
> "Flight #:\t921\t\tSeat:\t21C"
```

→ "Flight #: 921 Seat: 21C"

Adds a quotation mark but without ending the string too early.

```
> "Login Password:\t\t\"C3P0R2D2\""
```

→ "Login Password: "C3P0R2D2""



SPECIAL CHARACTERS INSIDE STRINGS

Some characters need backslash notation in JavaScript Strings

Places a backslash itself in the String

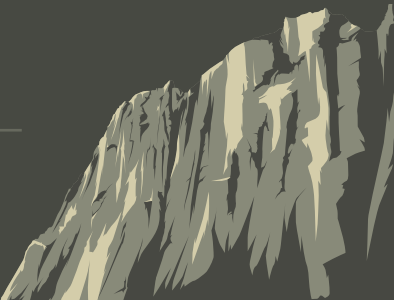
```
> "Origin\\Destination:\tOrlando(MCO)\\London(LHR)"
```

```
→ "Origin\\Destination:   Orlando(MCO)\\London(LHR)"
```

shifts the printout to a "new line"

```
> "Departure:\t09:55A\nArrival:\t14:55P"
```

```
→ "Departure: 09:55A  
→ Arrival:    14:55P"
```



STRING COMPARISONS

Checking for matching strings and alphabetical ordering

"Double equals" will compare EXACT contents

> "The Wright Brothers" == "The Wright Brothers"

→ true

> "The Wright Brothers" == "Super Mario Brothers"

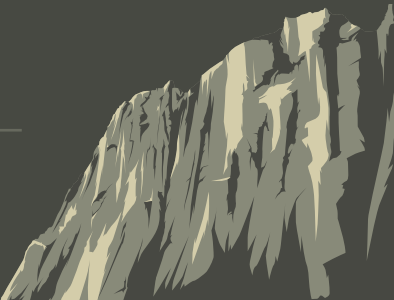
→ false

"Not equals" returns true if there is a mismatch

> "The Wright Brothers" != "the wright brothers"

→ true

Case counts!



STRING COMPARISONS

The length of strings can be accessed with the `.length` property

```
> "antidisestablishmentarianism".length
```

→ 28

Returns a number value

Spaces and any non-alphabetic characters are counted, too!

```
> "One Fish, Two Fish, Red Fish, Blue Fish".length
```

→ 39

