

PROTOTYPES AND INHERITANCE

**“AN OBJECT IS A SET OF KEY-VALUE PAIRS,
WHERE THE VALUES CAN ALSO BE FUNCTIONS”**

RECAP

“AN OBJECT IS A **SET OF KEY-VALUE PAIRS**,
WHERE THE VALUES CAN ALSO BE FUNCTIONS”

SO AN OBJECT HAS A BUNCH OF INFORMATION (**VALUES**), AND
YOU CAN ACCESS EACH BIT OF INFORMATION BY NAME (**KEYS**)

RECAP

“AN OBJECT IS A **SET OF KEY-VALUE PAIRS**,
WHERE THE VALUES CAN ALSO BE FUNCTIONS”

SO AN OBJECT HAS A BUNCH OF INFORMATION (VALUES), AND
YOU CAN ACCESS EACH BIT OF INFORMATION BY NAME (KEYS)

RECAP

“AN OBJECT IS A **SET OF KEY-VALUE PAIRS**,
WHERE THE VALUES CAN ALSO BE FUNCTIONS”

SO AN OBJECT HAS A BUNCH OF INFORMATION (VALUES), AND
YOU CAN ACCESS EACH BIT OF INFORMATION BY NAME (KEYS)

RECAP

“AN OBJECT IS A **SET OF KEY-VALUE PAIRS**,
WHERE THE VALUES CAN ALSO BE FUNCTIONS”

SO AN OBJECT HAS A BUNCH OF INFORMATION (VALUES), AND
YOU CAN ACCESS EACH BIT OF INFORMATION BY NAME (KEYS)

RECAP

“AN OBJECT IS A **SET OF KEY-VALUE PAIRS**,
WHERE THE VALUES CAN ALSO BE FUNCTIONS”

SO AN OBJECT HAS A BUNCH OF INFORMATION (VALUES), AND
YOU CAN ACCESS EACH BIT OF INFORMATION BY NAME (KEYS)

RECAP

**“AN OBJECT IS A SET OF KEY-VALUE PAIRS,
WHERE THE VALUES CAN ALSO BE FUNCTIONS”**

**EACH KEY-VALUE PAIR IS CALLED A PROPERTY
OF THE OBJECT**

RECAP

EACH KEY-VALUE PAIR IS CALLED A **PROPERTY**
OF THE OBJECT

REMEMBER THE TERM 'PROPERTY', ITS AN
IMPORTANT TERM!

RECAP

PROTOTYPES AND INHERITANCE

PROTOTYPES AND INHERITANCE

REMEMBER THAT JAVASCRIPT HAS NO

SAY YOU HAVE AN OBJECT CALLED **RECTANGLE**

```
function Rectangle(l,b){  
  this.length = l;  
  this.breadth = b;  
}
```

SO, WHEN WE SAY WE HAVE AN OBJECT CALLED RECTANGLE,

PROTOTYPES AND INHERITANCE

**REMEMBER THAT JAVASCRIPT
HAS NO CLASSES, ONLY OBJECTS -**

SAY YOU HAVE AN OBJECT CALLED



```
function Rectangle(l,b){  
  this.area=l*b;  
}
```

**SO, WHEN WE SAY WE HAVE AN OBJECT CALLED RECTANGLE, WE
MEAN WE HAVE AN OBJECT CONSTRUCTOR CALLED RECTANGLE**

PROTOTYPES AND INHERITANCE

REMEMBER THAT JAVASCRIPT HAS NO

SAY YOU HAVE AN OBJECT CALLED **RECTANGLE**

```
function Rectangle(l,b){  
  this.length = l;  
  this.breadth = b;  
}
```

SO, WHEN WE SAY WE HAVE AN OBJECT CALLED

SAY YOU HAVE AN OBJECT CALLED RECTANGLE

```
function Rectangle(l,b){  
    this.length = l;  
    this.breadth = b;  
}
```

SAY YOU HAVE AN OBJECT CALLED CIRCLE

```
function Circle(r) {  
    this.radius = r;  
}
```

SAY YOU HAVE AN OBJECT CALLED

NOW - SAY YOU WANTED TO ADD 2
PROPERTIES, SHAPETYPE, AND DRAW

WOULD YOU ADD THEM TO BOTH THE CIRCLE
AND THE RECTANGLE CLASS, I.E. TWICE?

IN ANY LANGUAGE - THE
ANSWER IS NO!

**NOW - SAY YOU WANTED TO ADD 2
PROPERTIES, SHAPETYPE, AND DRAW**

**WOULD YOU ADD THEM TO BOTH THE CIRCLE
AND THE RECTANGLE CLASS, I.E. TWICE?**

**IN ANY LANGUAGE - THE
ANSWER IS NO!**

**CIRCLE AND RECTANGLE ARE BOTH SHAPES,
PUT THE COMMON PROPERTIES INSIDE SHAPE**

SAY YOU HAVE AN OBJECT CALLED RECTANGLE

SAY YOU HAVE AN OBJECT CALLED CIRCLE

NOW - SAY YOU WANTED TO ADD 2
PROPERTIES, SHAPETYPE, AND DRAW

CIRCLE AND RECTANGLE ARE BOTH SHAPES,
PUT THE COMMON PROPERTIES INSIDE SHAPE

CIRCLE AND RECTANGLE ARE BOTH SHAPES,
PUT THE COMMON PROPERTIES INSIDE SHAPE

CREATE A SHAPE OBJECT CONSTRUCTOR, AND
SPECIFY THAT SHAPE IS THE **prototype**
FOR THE CIRCLE AND RECTANGLE OBJECTS

CREATE A SHAPE OBJECT CONSTRUCTOR, AND
SPECIFY THAT SHAPE IS THE prototype
FOR THE CIRCLE AND RECTANGLE OBJECTS

CREATE A SHAPE OBJECT CONSTRUCTOR, AND
SPECIFY THAT SHAPE IS THE **prototype**
FOR THE CIRCLE AND RECTANGLE OBJECTS

prototype IS A JAVASCRIPT KEYWORD, AND IS
A PROPERTY OF EVERY OBJECT CONSTRUCTOR

CREATE A SHAPE OBJECT CONSTRUCTOR, AND
SPECIFY THAT SHAPE IS THE **prototype**
FOR THE CIRCLE AND RECTANGLE OBJECTS

prototype IS A PROPERTY OF EVERY
OBJECT CONSTRUCTOR

HOW IS THIS POSSIBLE? IS N'T A CONSTRUCTOR
A FUNCTION?

ITS POSSIBLE - BECAUSE FUNCTIONS ARE
OBJECTS TOO!

SO ITS PERFECTLY OK FOR FUNCTIONS TO HAVE
PROPERTIES!

prototype IS A JAVASCRIPT KEYWORD, AND
IS A PROPERTY OF EVERY OBJECT CONSTRUCTOR

WHEN WE SAY THAT "SHAPE" IS THE **prototype** OF
"CIRCLE", IT MEANS THAT:

- EVERY OBJECT OF CIRCLE OR RECTANGLE WILL "HAVE ACCESS TO" 1 SPECIFIC OBJECT OF THE SHAPE CLASS
- THE PROPERTIES OF SHAPE WILL BE AVAILABLE ON EVERY CIRCLE OBJECT

prototype IS A JAVASCRIPT KEYWORD, AND
IS A PROPERTY OF EVERY OBJECT CONSTRUCTOR

PROTOTYPES ARE JAVASCRIPT'S WAY OF DOING INHERITANCE

- EVERY OBJECT OF CIRCLE OR RECTANGLE WILL HAVE ACCESS TO" 1 SPECIFIC OBJECT OF THE SHAPE CLASS
- THE PROPERTIES OF SHAPE WILL BE AVAILABLE ON EVERY CIRCLE OBJECT