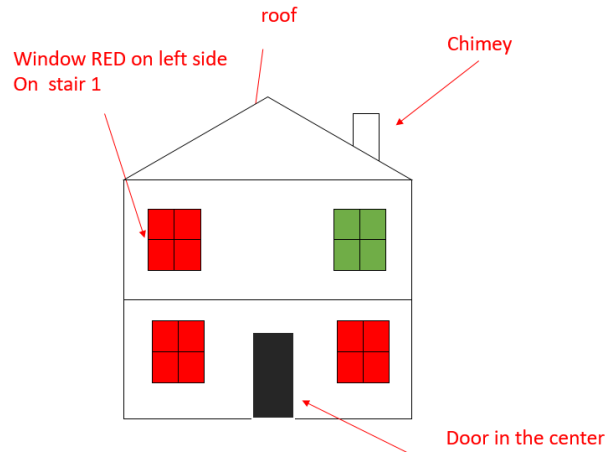


## C2- S1 HOUSE GAME



### NOTATION

To make the code more readable, we omit the constructor, so this code:

```
class Windows {  
  color: String;  
  stair : number;  
  side : string; // "LEFT" or "Right"  
}
```

} — **contructor** r

Is actually the following one *(including the constructor with color, stair, side as parameters)*

```
class Windows {  
  color: string;  
  stair : number;  
  side : string; // "LEFT" or "Right"  
  
  constructor (color:string, stair: number, side:string) {  
    this.color = color;  
    this.stair = stair;  
    this.side = side;  
  }  
}
```

## EXERCICE 1

Draw on paper how the house(s) should look like, given this code

### CLASSES

```
class House {  
  stairsNumber : number;  
  hasRoof : boolean;  
}
```

### OBJECTS

```
let myHouse = new House();  
myHouse.stairsNumber = 3;  
myHouse.hasRoof = true;
```

## EXERCICE 2

Draw on paper how the house(s) should look like, given this code

### CLASSES

```
class House {  
  stairsNumber : number;  
  hasRoof : boolean = false;  
}
```

### OBJECTS


```
let myHouse = new House();  
myHouse.stairsNumber = 2;
```

### EXERCICE 3

Draw on paper how the house(s) should look like, given this code

#### CLASSES

```
class House {  
  stairsNumber : number = 2;  
  hasRoof : boolean = true;  
  windows : Windows[]  
}  
class Windows {  
  color: String;  
  stair : number;  
  side : string; // "LEFT" or "Right"  
}
```



#### OBJECTS

```
let w1 = new Windows("BLUE", 0, "LEFT");  
let w2 = new Windows("RED", 1, "RIGHT");  
let w3 = new Windows("BLUE", 1, "LEFT");  
  
let myHouse = new House();  
myHouse.windows = [w1, w2, w3];
```

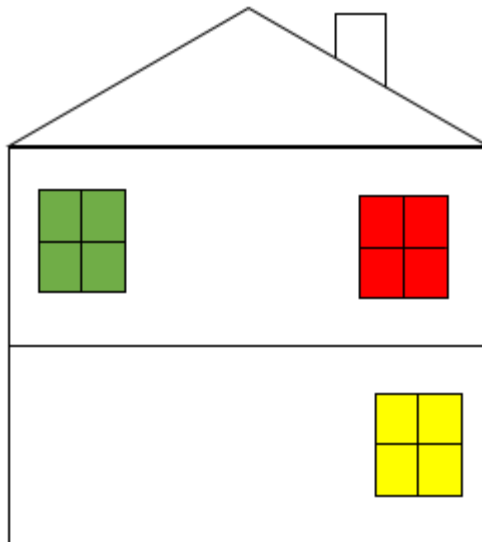
## EXERCICE 4

Write the code to instantiate object to create a house like the below draw

### CLASSES

```
class House {  
    stairsNumber : number = 2;  
    hasRoof : boolean = true;  
    windows : Windows[]  
}  
class Windows {  
    color: String;  
    stair : number;  
    side : string; // "LEFT" or "Right"  
}
```

} constructor




## EXERCICE 5

Draw on paper how the house(s) should look like, given this code

### CLASSES

```
class House {  
  stairsNumber : number = 2;  
  hasRoof : boolean = true;  
  windows : Windows[]  
}  
class Windows {  
  color: String;  
  stair : number;  
  side : string; // "LEFT" or "Right"  
}
```



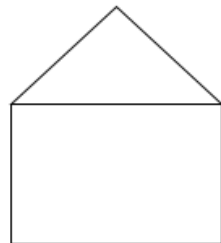
### OBJECTS

```
let w1 = new Windows("GREEN", 0, "LEFT");  
let w2 = new Windows("RED", 1, "LEFT");  
let w3 = new Windows("BLUE", 1, "LEFT");  
  
let house1 = new House();  
house1.windows = [w3];  
house1.hasRoof = false;  
  
let house2 = new House();  
house2.windows = [w1, w2];
```

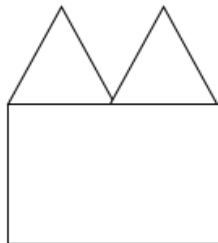
## EXERCICE 6

Now we define 3 kind of roofs, defined by their type (string)

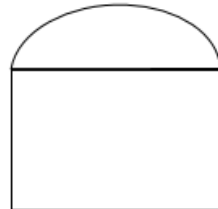
### ROOF TYPES



V SHAPE



M SHAPE



U SHAPE

Draw on paper how the house(s) should look like, given this code

### CLASSES

```
class House {  
  stairsNumber : number = 2;  
  roof : Roof;  
  windows : Windows[]  
}
```

```
class Windows {  
  color: String;  
  stair : number;  
  side : string; // "LEFT" or "Right"  
}
```

} constructor

```
class Roof {  
  // "V SHAPE" or "M SHAPE" or "U SHAPE"  
  style: String;  
  hasChimney : boolean ;  
}
```

} constructor

### OBJECTS

```
let w1 = new Windows("BLUE", 1, "LEFT");  
let myRoof = new Roof("M SHAPE" , true);
```

```
let myHouse = new House();  
myHouse.windows = [w1];  
myHouse.roof = myRoof;
```

## EXERCICE 7

Draw on paper how the house(s) should look like, given this code

### CLASSES

```
class House {  
    stairsNumber : number;  
    roof : Roof;  
    windows : Windows[]  
    mainDoor : Door  
}  
  
class Windows {  
    color: String;  
    stair : number;  
    side : string; // "LEFT" or "Right"  
}  
  
class Roof {  
    // "V SHAPE" or "M SHAPE" or "U SHAPE"  
    style: String;  
    hasChimney : boolean = true;  
}  
  
class Door {  
    // ""LEFT" or "Right" or "CENTER"  
    position: String;  
}
```

} constructor

} constructor

} constructor

} constructor

```
let w1 = new Windows("BLUE", 1, "LEFT");  
let w2 = new Windows("BLUE", 1, "RIGHT");  
  
let myHouse = new House();  
  
let myRoof = new Roof("U SHAPE");  
let myDoor = new Door("LEFT");  
  
myHouse.roof = myRoof;  
myHouse.windows = [w1, w2];  
myHouse.door = myDoor;
```



## EXERCICE 8

Write the code to instantiate object to create a house like the below draw

### CLASSES

```
class House {  
    stairsNumber : number;  
    roof : Roof;  
    windows : Windows[]  
    mainDoor : Door  
}  
  
class Windows {  
    color: String;  
    stair : number;  
    side : string; // "LEFT" or "Right"  
}  
  
class Roof {  
    // "V SHAPE" or "M SHAPE" or "U SHAPE"  
    style: String;  
    hasChimney : boolean = true;  
}  
  
class Door {  
    // ""LEFT" or "Right" or "CENTER"  
    position: String;  
}
```

} contructor

} contructor

} contructor

} contructor

