

Project 65

65



Coding School



- ❖ Let's design a shop to buy goods.
- ❖ The types of goods to buy in the shop and the images needed to design the shop are given to you in the library. Let's see how to design using them as follows.



- ❖ Let's use the code blocks used to create a sprite to add the image to the shop.

```
var shop = createSprite(200, 210, 500, 700);
shop.setAnimation(▼ "shop.png");
shop.scale = 1.2;
```

Create the sprite as Shop and give its x and y positions as 200 and 210.

Use the “setAnimation” block to set the animation for the sprite. For that, select the image of the shop which is "shop.png".

Give the scale of the sprite as 1.2.

- ❖ After designing the shop, let's design the cart.

- ❖ Create a sprite for the cart. Select "cart.png" as its animation in setAnimation. Its positions are as follows.

X – 200

Y – 360

Scale – 0.5 Use these values to create the cart.

- ❖ Then let's design the items on the shelves of the shop.

```
var cart = createSprite(200, 360);
cart.setAnimation("cart.png");
cart.scale = 0.5;
```

- ❖ Arrange the blocks as follows to create the carrot on the corresponding shelf.

Create a sprite and name it "carrots". For that, set the "carrots" image using setAnimation as above. Give its x and y position as "70,70"..

Set the scale of the Carrots sprite to 0.3.

```
var carrots = createSprite(70, 70);
carrots.setAnimation("carrots.png");
carrots.scale = 0.3;
```

- ❖ By creating new sprites in the same way, correctly select the given animation and place other types of items in the appropriate places on the shelf. Blocks related to beans are given below.

```
var green_beans = createSprite(200, 70, 300);
green_beans.setAnimation("green_beans.png");
green_beans.scale = 0.3;
```

- ❖ Here, gives the correct x, y positions to place the goods on the shelves of the shop. Give the same as above to fit the scale.
- ❖ After designing all the sprites to create them on the screen in the draw function drawSprites(); Call the function.

```
function draw() {
  drawSprites();
}
```

- ❖ At the beginning of coding, create two variables as price and total as follows.

```
var total = 0;  
var price = 0;
```

- ❖ Then click on each item and enter the relevant codes to put it in the cart.
- ❖ Add Carrots to cart and add codes below.

```
if (mousePressedOver(carrots)) {  
  carrots.x = World.mouseX;  
  carrots.y = World.mouseY;  
  price = 220;  
}
```

- ❖ Here, when you click on the carrots and drag them on the screen with the mouse, these codes are used

```
carrots.x = World.mouseX;
```

 to move the carrots along with them.

```
carrots.y = World.mouseY;
```

- ❖ Let's compare the price of the selected item to the price variable to know the price of the selected item. By this means it is possible to get the complete collection

```
price = 220;
```

 of goods purchased.

- ❖ Since the price of carrots is 220, let's make the price variable equal to 220 as above.

- ❖ In the same way, use the if block and create codes related to adding other items to the cart.

- ❖ The codes related to beans are as follows. where the price variable is set equal to the value of beans.

```
if (mousePressedOver(green_beans)) {  
  green_beans.x = World.mouseX;  
  green_beans.y = World.mouseY;  
  price = 350;  
}
```

- ❖ Then let's code what should happen when the items are put in the cart.
- ❖ If carrots are put in the cart, when they touch the cart, it should disappear and its amount should be added to the total.

- ❖ For that, use the is touching block in an if block and code what should happen if the cart is touched by the carrot.

```
if ( carrots.isTouching(cart) ) {
  total = total+price;
  carrots.destroy();
}
```

- ❖ What happens is that the price of the item newly added to the cart is added to the total of the purchased items and the related items are lost.
- ❖ The above blocks indicate that when the carrot is put in the cart, its value is added to the total and the carrot is lost.
- ❖ In the same way, create codes related to what happens when other types of goods are added to the cart.
- ❖ The codes related to beans are given below.

```
if ( green_beans.isTouching(cart) ) {
  total = total+price;
  green_beans.destroy();
}
```

- ❖ Then let's code as below to show the price related to all purchased items.

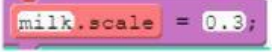
```
textSize(20);
fill("black");
text("Total: " + total + "/=", 20, 350);
```

- ❖ The textSize block can change the size of the displayed text.
- ❖ Fill can change its color. Black colour is used here.
- ❖

```
text("Total: " + total + "/=", 20, 350);
```

 This will display a text as Total : and after that the value of the total variable will be displayed and at the end /= sign can be created. Ex: Total: 220/=
- ❖ Its position is x-20, y-350.
- ❖ Do all these codes in the draw function.

- Select The correct answer.

1.  By increasing the value of this the respective sprite,

Can be enlarged.

Can be made smaller.

No change in size

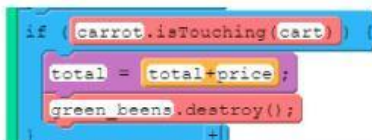
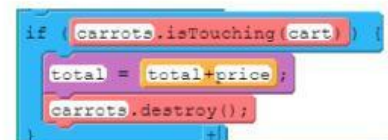
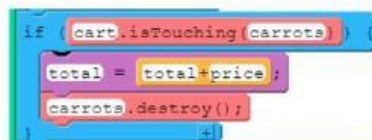
2.  Here,

The value of the total is added to the total.

Total is equal to total or price.

The value of the price is added to the total

3. What should happen when adding goods to the cart from the blocks given below, what is the wrong block set?



4. What is the correct code to change the color of the text showing the total to red?

```
fill(▼ "black");  
fill(▼ red);
```

```
fill(rgb(255, 0, 0));
```

5.

```
text("Total: " + total + "/", 20, 350);
```

“Rs.” in front of the price for the part displaying this price. What is the wrong code when entering as?

```
text("Total: Rs." + total + "/", 20, 350);
```

```
text("Total: "+"Rs." + total + "/", 20, 350);
```

```
text(("Total: " + "Rs.") + total + "/", 20, 350);
```