

# Computing with Events

CMSC 326 Simulations

# Computing with Events

Coin Flips



Dice



Cards



# Computing with Events

Coin Flips



Dice



Cards



## Coin Flips

```
# Flip a coin 5 times.  
coin = Coin()  
for i in range(5):  
    # Returns 1 (heads) or 0 (tails).  
    flip_result = coin.flip()  
    print(f"Flip #{i}: {flip_result}")
```

# Coin Flips

## Run 1

```
Flip #0: 1  
Flip #1: 1  
Flip #2: 0  
Flip #3: 0  
Flip #4: 1
```

## Run 2

```
Flip #0: 0  
Flip #1: 1  
Flip #2: 0  
Flip #3: 0  
Flip #4: 1
```

## Run 3

```
Flip #0: 1  
Flip #1: 1  
Flip #2: 0  
Flip #3: 1  
Flip #4: 0
```

# Coin Flips

```
# A simple coin
class Coin:
    def __init__(self, probability_of_heads=0.5):
        self.p = probability_of_heads

    def flip(self):
        # Return 1 (heads) or 0 (tails).
        if random.random() < self.p:
            return 1
        else:
            return 0
```

# Coin Flips

```
# A simple coin
```

```
class Coin:
```

```
    def __init__(self, probability_of_heads=0.5):  
        self.p = probability_of_heads
```

```
    def flip(self):
```

```
        # Return 1 (heads) or 0 (tails).
```

```
        if random.random() < self.p:
```

```
            return 1
```

```
        else:
```

```
            return 0
```

# Coin Flips

```
# A simple coin
class Coin:
    def __init__(self, probability_of_heads=0.5):
        self.p = probability_of_heads

    def flip(self):
        # Return 1 (heads) or 0 (tails).
        if random.random() < self.p:
            return 1
        else:
            return 0
```



# Coin Flips

`random.random( )`

Return the next random floating-point number in the range  $0.0 \leq X < 1.0$

# Coin Flips

