

# Looping



# Three Doors Game - The Monty Hall Problem

[Video](#)



Is there any advantage or disadvantage to switching?

# Find the Disney Star in this video?

- Green ticket to the first with the right answer...
- Make sure you use BQ in the subject
- [Video](#)
- Movie clip from '21'



# Looping Statements: Doing something more than once

- While loop
- Do-While loop
- For loop



# While loop syntax


```
while (<Boolean expression>)  
{  
    <statements>  
}
```



# While Loop Example

```
int count = 0;
const int max = 5;
int sum = 0;

while (count < max)
{
    sum = sum + count;
    count = count + 1;
}
labelAnswer.Text = Convert.ToString(sum);
```



## Do-While loop syntax:

```
do
{
    <statements>
} while (<Boolean expression>);
```



# Do-While Loop Example

```
int count = 0;
const int max = 5;
int sum = 0;

do
{
    sum = sum + count;
    count = count + 1;
} while (count < max);
labelAnswer.Text = Convert.ToString(sum);
```





## For loop syntax:

```
for (<initialization statement>;  
    <Boolean expression>;  
    <iterator statement>)  
{  
    <statements>  
}
```



# For Loop Example

```
const int max = 5;
int sum = 0;


for (int count = 0; count < max; count++)
{
    sum = sum + count;
}
labelAnswer.Text = Convert.ToString(sum);
```



# More Examples


```
int count = 1;
const int max = 100;
int sum = 0;

while (sum < max)
{
    sum = sum + count;
    count = count + 1;
}
labelAnswer.Text = Convert.ToString(count);
```



# More Examples

```
Random random = new Random();
int count = 0;
int diceValue1;
int diceValue2;
do
{
    diceValue1 = random.Next(1, 7);
    diceValue2 = random.Next(1, 7);
    count = count + 1;
} while (diceValue1 != diceValue2);
labelAnswer.Text = Convert.ToString(count);
```



# But Wait! There's More!!!

- break;
- continue;



# Break Statement: How to get out early

- Break syntax:  
`break;`



# Continue Statement: How to skip this one

- Continue syntax:  
`continue;`



# Looping Assignment

The screenshot shows a Windows application window titled "Looping Assignment". It contains three distinct panels for different programming exercises:

- X to the Nth Power:** This panel has two input fields. The first is labeled "X:" and contains the value "3". The second is labeled "N:" and contains the value "2". Below these fields is a "Calculate" button. The output area below the button displays "Answer: 9".
- Loan Calculator:** This panel has two input fields. The first is labeled "Loan amount:" and contains "10000". The second is labeled "Monthly payment:" and contains "250". Below these fields is a "Calculate" button. The output area below the button displays "Loan paid off in 40 months."
- Hailstone Sequence:** This panel has one input field labeled "Starting value:" containing "11". Below it is a "Start Sequence" button. The output area below the button displays the sequence of numbers: "11 34 17 52 26 13 40 20 10 5 16 8" on the first line and "4 2 1" on the second line.



