

# Advanced Functions

## Anonymous functions

- `funchandle = @(arguments) functionbody;`

```
>> circlearea = @(radius) pi*radius^2;
```

```
>> circlearea(4)
```

```
ans =  
    50.2655
```

```
>> dieroll = @( ) randi(6);
```

```
>> dieroll
```

```
dieroll =  
    @( )randi(6)
```

```
>> dieroll( )
```

```
ans =  
     4
```

## Function handles

- Placing a '@' sign in front of a function name makes it a function handle.
- Exercise: Write a function `trigplot(func)` that takes a function handle `func` and plots it for `[0:1:2*pi]`.

## Conversion to/from strings

- `funchandle = str2func ( str )`
- `str = func2str ( funchandle )`
- Exercise: Change the `trigplot` function so it also accepts the function to plot as a string.

# Nested Functions

```
outer function header

    body of outer function

    inner function header
        body of inner function
    end % inner function

    more body of outer function

end % outer function
```

- Warning: Nested functions share the same workspace.

# Persistent Variables

- persistent variablename
- isempty(variablename)
- Exercise: Write a function countme() that returns the number of times it has been called.

# Global Variables (Ahmet does not recommend using global variables)

- Used when you need to share variables between functions, without having to input/return them.
- global variablename
- Exercise: Write a function givetwo() that increases a global variable mytwocents by 2. Write another function taketwo() that decreases it by 2.