

# Three fundamental concepts

(plus a little borrowed magic)

## Expressions

- expressions: have a *value* of a specific *type* (alternatively: they belong to a specific *class*)
- name values with the `val` keyword

## Examples

```
val hmt = "Homer Multitext"  
val venetusAPageCount = 327  
val areGreekManuscriptsAwesome = true
```

## Functions

- create (*return*) a value
- classes (types) can have functions defined for all instances of that class
  - syntax: *value.function*

## Examples

`size` is a function on Strings.

`"Homer Multitext".size`

what class of value does it return?

## Parameters

Functions can take *parameters* (*arguments*) in parentheses

## Example

`replaceAll` is a function on strings that takes two parameters.

```
"The HMT project".replaceAll("HMT", "Homer Multitext")
```

## Code libraries

Bring other packages into your script with `import` keyword



## Example

```
import edu.holycross.shot.cite._
```

(Note the `_` wildcard to import *all* classes in this package.)

This doesn't do anything directly, but makes the imported classes available to your script.

## A class for CTS URNs

```
val iliad = CtsUrn("urn:cts:greekLit:tlg0012.tlg001:")  
val iliad20 = CtsUrn("urn:cts:greekLit:tlg0012.tlg001:20")
```

- What type are they?
- Find out some functions on them with tab completion

First URN magic

Try this:

```
iliad ~~ iliad20
```

Proceed to Scala collections

