João Gonçalves

Section 3

Pattern Matching





### In this Section, we are going to take a look at...

- Fundaments of pattern matching
- Difference from assignment
- Forms of pattern matching
- Practical use cases



João Gonçalves

Video 3.1
Pattern Matching
versus Assignment



#### In this video, we are going to take a look at...

- What is pattern matching
- How does it differ from assignment

vind 5s















# list = [1,2,3,4,5]



$$list = [1,2,3,4,5]$$

Assign the value of the list to the variable named list



## x = 1

We can do this with any type



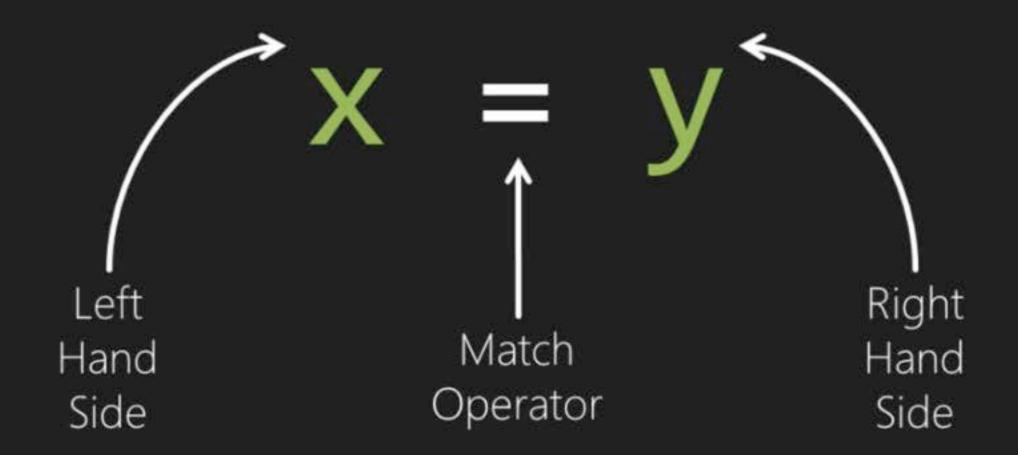
x = 1 1 = x



# x = 1 1 = x 1 = 1









x = y

Match the left hand side with the value on the right hand side





#### **Terminal**

```
iex(1) > x = [1,2,3,4,5]
[1,2,3,4,5]
iex(2) > x
[1,2,3,4,5]
iex(3) > [1,2,3,4,5] = x
[1,2,3,4,5]
```





#### **Terminal**

```
iex(4)> x = 3
3
iex(5)> x
3
iex(6)> [1,2,3,4,5] = x
** (MatchError) no match of right hand side value: 3
```



### Matching Rules



If the left hand side contains a variable, the variable on the right hand side is bound to the variable



#### Matching Rules

If the right hand side contains a name, the left hand side is matched to the value of the variable with that name or a function with the same name, if it exists



### Matching Rules

$$x = 1$$
  
 $x = 2$ 

$$x = 2$$

A variable can be "re-assigned" with a different value on a subsequent match



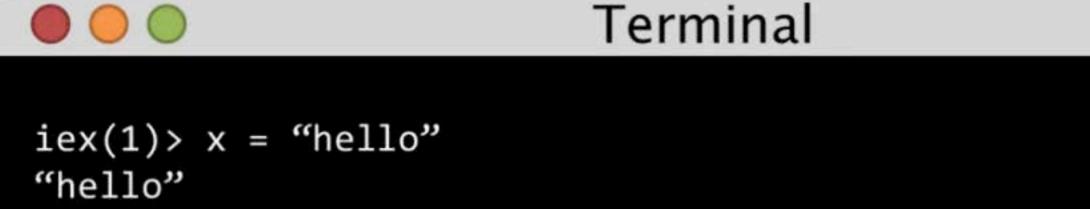
$$\frac{\text{Pin}}{\text{Operator}} \xrightarrow{} \bigwedge X = Y$$



$$\frac{\text{Pin}}{\text{Operator}} \xrightarrow{\bigwedge} \frac{\bigwedge}{X} = \frac{1}{2}$$

Strict check for a match, no binding of variables





iex(2)> x = "hey"
"hey"
iex(3)> ^x = "oi"

\*\* (MatchError) no match of right hand side value: "oi"





#### **Terminal**

```
iex(4)> ^y = "hello"
** (CompileError) iex:4: unbound variable ^y
```

