Structures

Grouping related information

Structures

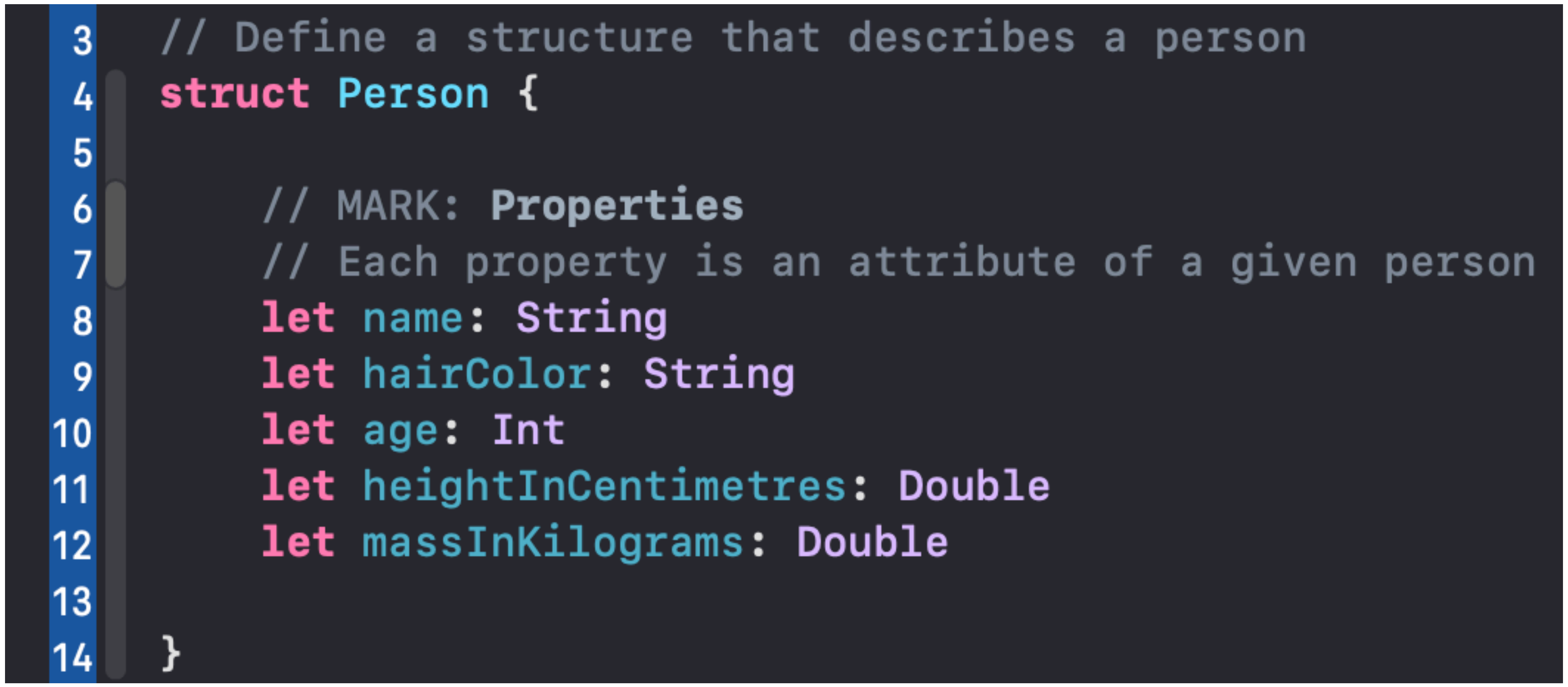
Grouping related information

- Individual variables or constants only go so far
- What if we want to describe, say, a person?
- What attributes does a *person* have?
    - Name
    - Hair colour
    - Age
    - Height
    - Mass

Structures

Grouping related information

- Let's examine this.
    - Add a *Structures* playground to your *Notes* project.
- The structure would look like this:

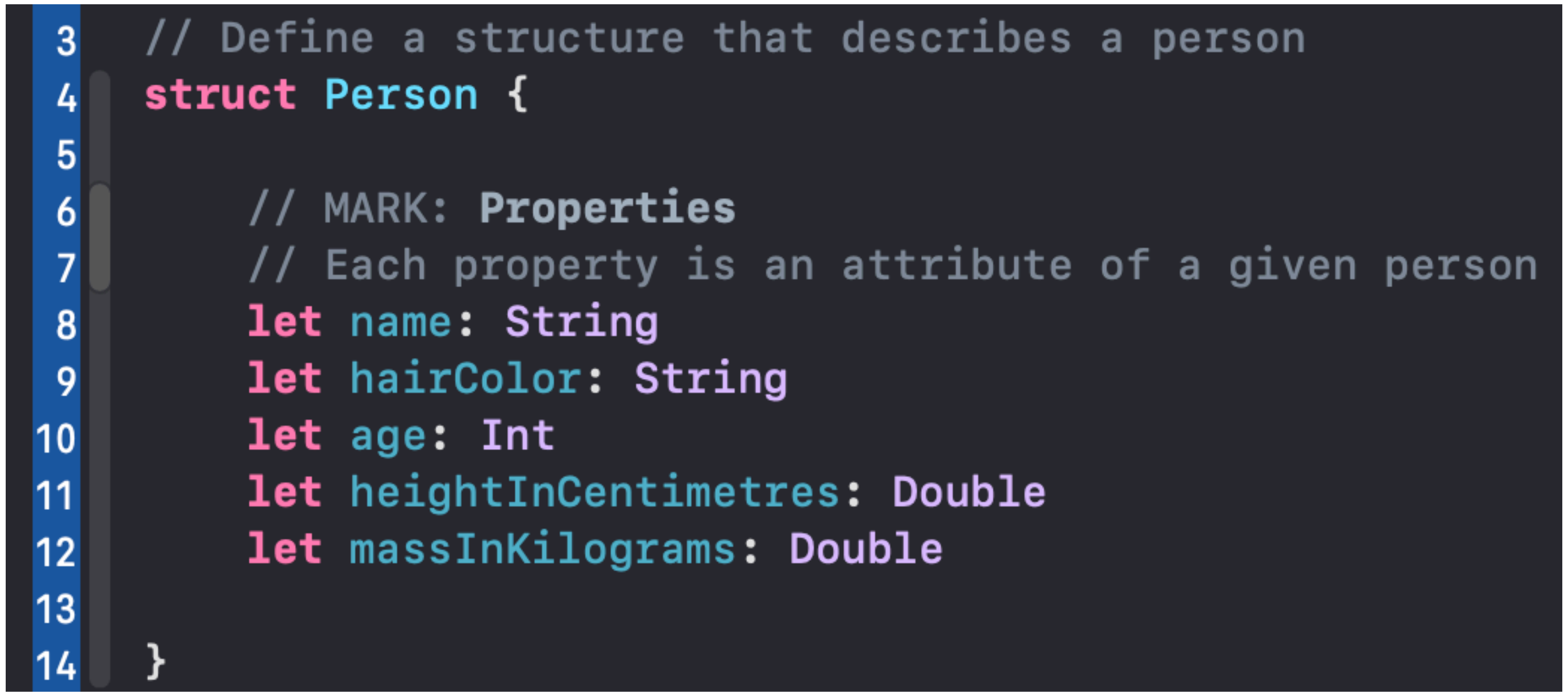


```swift
// Define a structure that describes a person
struct Person {

    // MARK: Properties
    // Each property is an attribute of a given person
    let name: String
    let hairColor: String
    let age: Int
    let heightInCentimetres: Double
    let massInKilograms: Double

}
```

Structures

Grouping related information

- This is the *definition* of a structure.



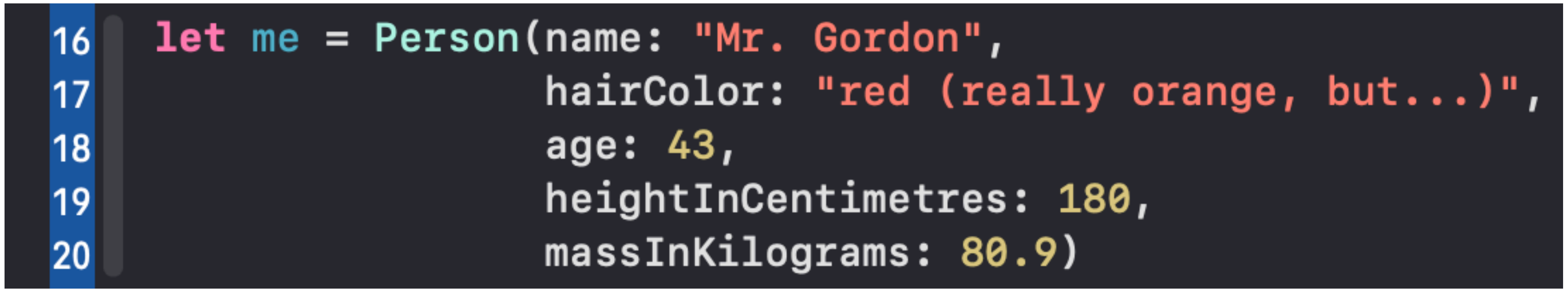
```swift
// Define a structure that describes a person
struct Person {

    // MARK: Properties
    // Each property is an attribute of a given person
    let name: String
    let hairColor: String
    let age: Int
    let heightInCentimetres: Double
    let massInKilograms: Double

}
```

- But how do we *use* a structure?

Structures

Grouping related information

- A structure is a *type*.
- We are literally creating a new data type in Swift!
- To create an instance of a type, we must provide values for each property of the type

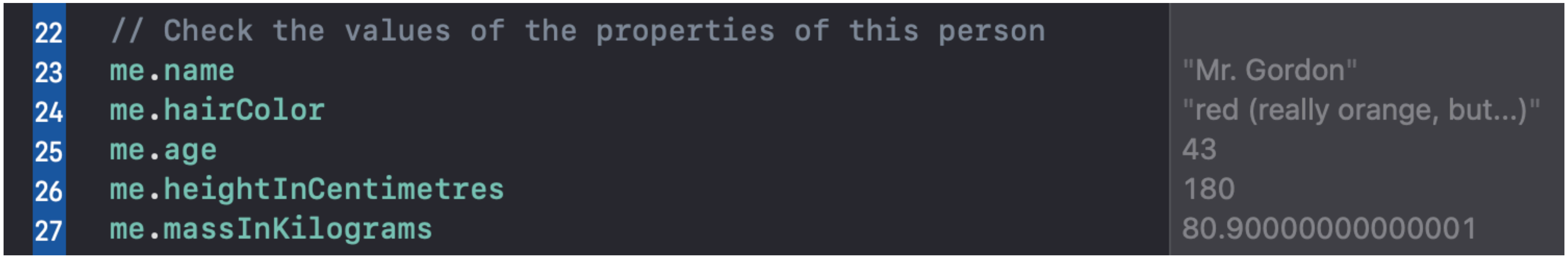


```swift
let me = Person(name: "Mr. Gordon",
                hairColor: "red (really orange, but...)",
                age: 43,
                heightInCentimetres: 180,
                massInKilograms: 80.9)
```

Structures

Grouping related information

- Once an *instance* of the type is created, we can query it.

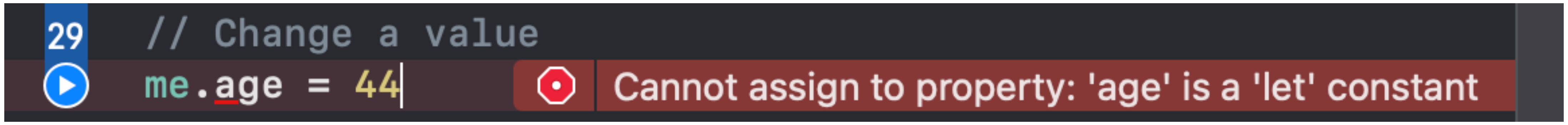


```
// Check the values of the properties of this person
me.name                          "Mr. Gordon"
me.hairColor                     "red (really orange, but...)"
me.age                           43
me.heightInCentimetres           180
me.massInKilograms               80.90000000000001
```

Structures

Grouping related information

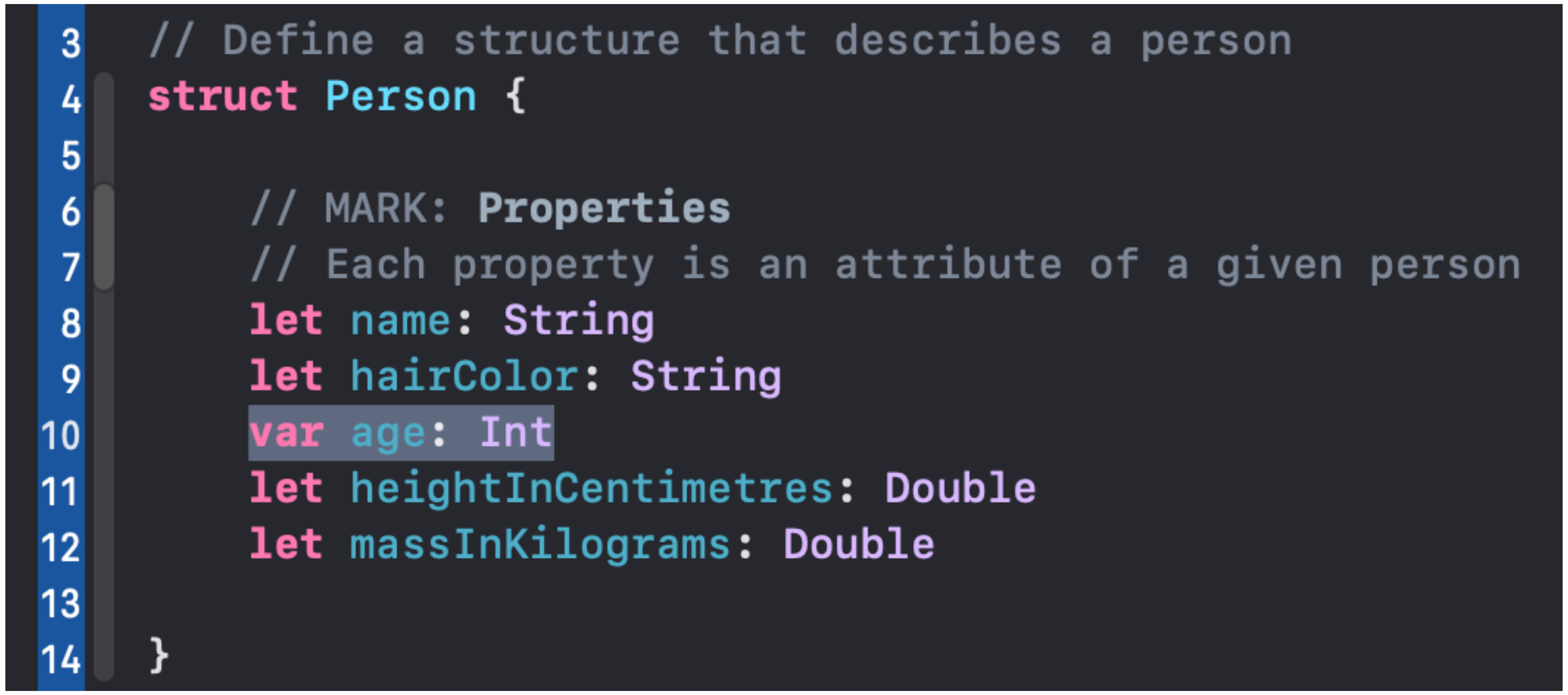
- What happens if we try to change one of these values?



```swift
// Change a value
me.age = 44
```

Cannot assign to property: 'age' is a 'let' constant

- OK, let's adjust the property:

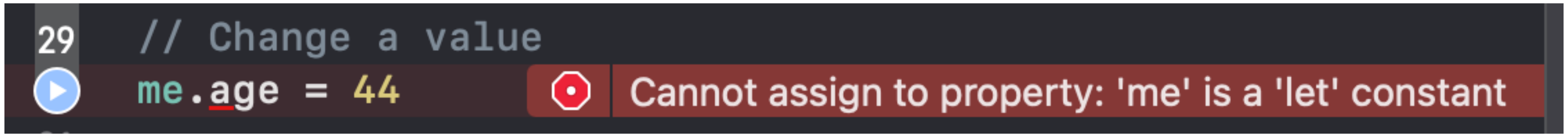


```swift
// Define a structure that describes a person
struct Person {

    // MARK: Properties
    // Each property is an attribute of a given person
    let name: String
    let hairColor: String
    var age: Int
    let heightInCentimetres: Double
    let massInKilograms: Double

}
```

Structures

Grouping related information

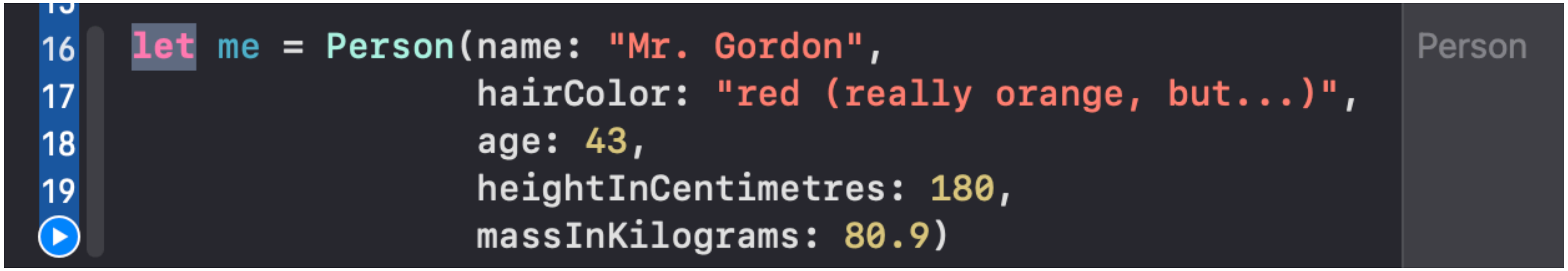
- Now let's try again...



```
// Change a value
me.age = 44
```

Cannot assign to property: 'me' is a 'let' constant

- Ah... remember, we made the *me* instance a constant.



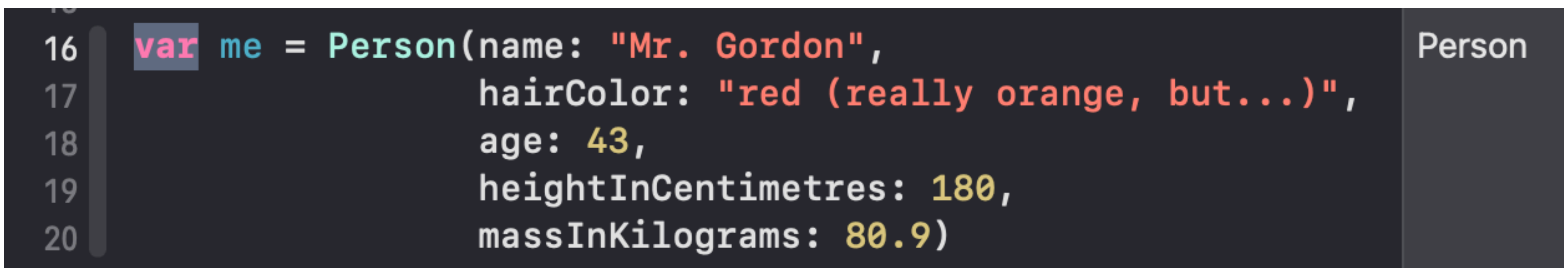
```
let me = Person(name: "Mr. Gordon",
                hairColor: "red (really orange, but...)",
                age: 43,
                heightInCentimetres: 180,
                massInKilograms: 80.9)
```

Person

Structures

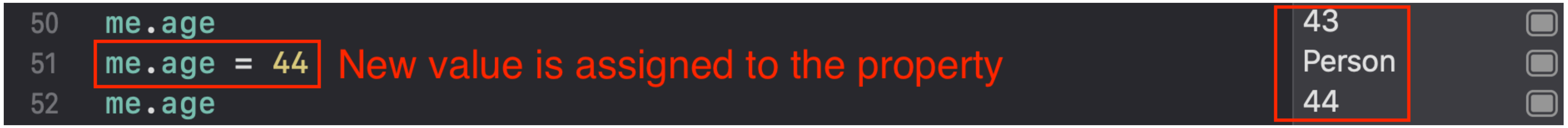
Grouping related information

- Let's adjust that...



```
var me = Person(name: "Mr. Gordon",
                hairColor: "red (really orange, but...)",
                age: 43,
                heightInCentimetres: 180,
                massInKilograms: 80.9)
```

- Now we can change the age:



```
me.age
me.age = 44
me.age
```

- So... structures work like any other type.
    - Declared as a constant, cannot be changed.
    - Declared as a variable... and properties that are also variables can change.

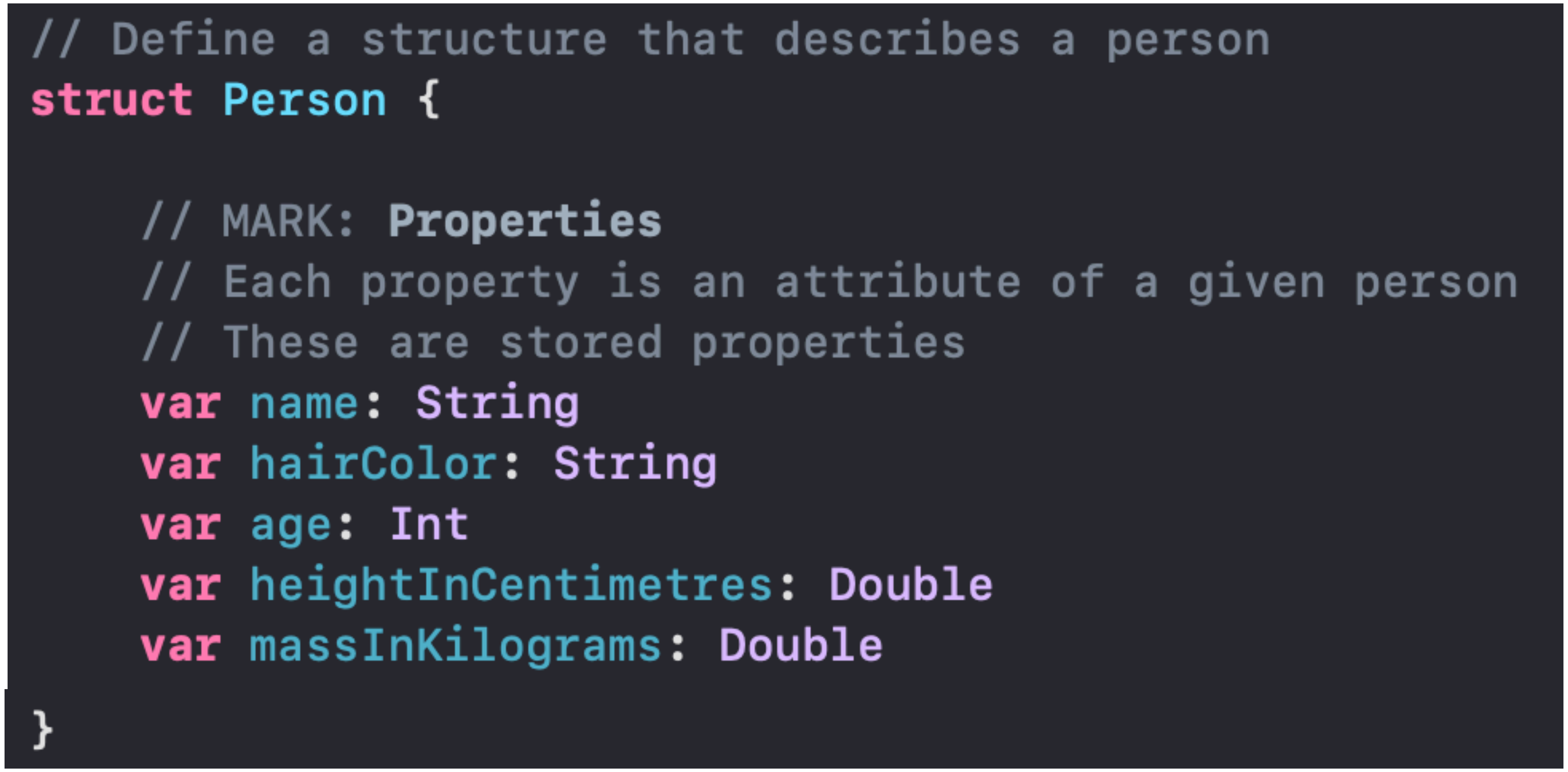
Structures

Grouping related information

- Mini-exercise
  - Discuss with your neighbour
  - What other properties should be variables?
  - Make any necessary changes to the Person structure.
    - Of course, all the properties of this structure describe attributes that could potentially change in a person's lifetime
    - So... see next page

Structures

Grouping related information



```swift
// Define a structure that describes a person
struct Person {

    // MARK: Properties
    // Each property is an attribute of a given person
    // These are stored properties
    var name: String
    var hairColor: String
    var age: Int
    var heightInCentimetres: Double
    var massInKilograms: Double

}
```