

Enumeration

Enumeration

Common type for a group of related values

Type safe environment

Can have raw and associated values

Great integration with switch conditional

enum

```
// Declaring an enumeration
```

```
enum Country {  
    case indonesia  
    case singapore  
    case malaysia  
    case brunei  
    case philippines  
    case eastTimor  
}
```

```
// Accessing the case
```

```
let universityCountry = Country.indonesia
```

```
// Presenting the value
```

```
print( universityCountry )
```

```
// Declaring an enumeration
```

```
enum Country {
```

```
    case indonesia
```

```
    case singapore
```

```
    case malaysia
```

```
    case brunei
```

```
    case philippines
```

```
    case eastTimor
```

```
}
```

```
// Accessing the case
```

```
let universityCountry = Country.indonesia
```

```
// Presenting the value
```

```
print( universityCountry )
```

```
// Naming as a Type
```

```
enum Country {  
    case indonesia  
    case singapore  
    case malaysia  
    case brunei  
    case philippines  
    case eastTimor  
}
```

```
// Accessing the case
```

```
let universityCountry = Country.indonesia
```

```
// Presenting the value
```

```
print( universityCountry )
```

```
// Naming as a Type
```

```
enum Country {
```

```
    case indonesia
```

```
    case singapore
```

```
    case malaysia
```

```
    case brunei
```

```
    case philippines
```

```
    case eastTimor
```

```
}
```

```
// Accessing the case
```

```
let universityCountry = Country.indonesia
```

```
// Presenting the value
```

```
print( universityCountry )
```

```
// Naming as a Type
```

```
enum Country {
```

```
    case indonesia
```

```
    case singapore
```

```
    case malaysia
```

```
    case brunei
```

```
    case philippines
```

```
    case eastTimor
```

```
}
```

```
// Accessing the case
```

```
let universityCountry = Country.indonesia
```

```
// Presenting the value
```

```
print( universityCountry )
```

```
// Naming as a Type
```

```
enum Country {  
    case indonesia  
    case singapore  
    case malaysia  
    case brunei  
    case philippines  
    case eastTimor  
}
```

```
// Accessing the case
```

```
let universityCountry = Country.indonesia
```

```
// Presenting the value
```

```
print( universityCountry )
```



```
// Naming as a Type
```

```
enum Country {  
    case indonesia  
    case singapore  
    case malaysia  
    case brunei  
    case philippines  
    case eastTimor  
}
```

```
// Accessing the case
```

```
let universityCountry = Country.indonesia
```

```
// Presenting the value
```

```
print( universityCountry )
```

```
// Naming as a Type
```

```
enum Country {  
    case indonesia  
    case singapore  
    case malaysia  
    case brunei  
    case philippines  
    case eastTimor  
}
```

```
// Accessing the case
```

```
let universityCountry = Country.indonesia
```

```
// Presenting the value
```

```
print( universityCountry )
```

```
// Naming as a Type
```

```
enum Country {  
    case indonesia  
    case singapore  
    case malaysia  
    case brunei  
    case philippines  
    case eastTimor  
}
```

```
// Accessing the case
```

```
let universityCountry = Country.indonesia
```

```
// Presenting the value
```

```
print( universityCountry )
```

indonesia

```
enum Move {  
    case up, down, right, left  
}  
  
let characterNextMove: Move = .right  
  
switch characterNextMove {  
    case Move.up:  
        print("Hero will jump")  
    case Move.down:  
        print("Hero will crouch")  
    case .right:  
        print("Hero will walk")  
    case .left:  
        print("Hero will block")  
}
```

```
enum Move {  
    case up, down, right, left  
}
```

```
let characterNextMove: Move = .right
```

```
switch characterNextMove {  
    case Move.up:  
        print("Hero will jump")  
    case Move.down:  
        print("Hero will crouch")  
    case .right:  
        print("Hero will walk")  
    case .left:  
        print("Hero will block")  
}
```

```
enum Move {  
    case up, down, right, left  
}
```

```
let characterNextMove: Move = .right
```

```
switch characterNextMove {  
    case Move.up:  
        print("Hero will jump")  
    case Move.down:  
        print("Hero will crouch")  
    case .right:  
        print("Hero will walk")  
    case .left:  
        print("Hero will block")  
}
```

```
enum Move {  
    case up, down, right, left  
}  
  
let characterNextMove: Move = .right  
  
switch characterNextMove {  
    case Move.up:  
        print("Hero will jump")  
    case Move.down:  
        print("Hero will crouch")  
    case .right:  
        print("Hero will walk")  
    case .left:  
        print("Hero will block")  
}
```

```
enum Move {  
    case up, down, right, left  
}  
  
let characterNextMove: Move = .right  
  
switch characterNextMove {  
    case Move.up:  
        print("Hero will jump")  
    case Move.down:  
        print("Hero will crouch")  
    case .right:  
        print("Hero will walk")  
    case .left:  
        print("Hero will block")  
}
```



```
enum Move {  
    case up, down, right, left  
}  
  
let characterNextMove: Move = .right  
  
switch characterNextMove {  
    case Move.up:  
        print("Hero will jump")  
    case Move.down:  
        print("Hero will crouch")  
    case .right:  
        print("Hero will walk")  
    case .left:  
        print("Hero will block")  
}
```

```
enum Move {  
    case up, down, right, left  
}  
  
let characterNextMove: Move = .right  
  
switch characterNextMove {  
    case Move.up:  
        print("Hero will jump")  
    case Move.down:  
        print("Hero will crouch")  
    case .right:  
        print("Hero will walk")  
    case .left:  
        print("Hero will block")  
}
```

```
enum Move {  
    case up, down, right, left  
}  
  
let characterNextMove: Move = .right  
  
switch characterNextMove {  
    case Move.up:  
        print("Hero will jump")  
    case Move.down:  
        print("Hero will crouch")  
    case .right:  
        print("Hero will walk")  
    case .left:  
        print("Hero will block")  
}
```

```
enum Move {  
    case up, down, right, left  
}  
  
let characterNextMove: Move = .right  
  
switch characterNextMove {  
    case Move.up:  
        print("Hero will jump")  
    case Move.down:  
        print("Hero will crouch")  
    case .right:  
        print("Hero will walk")  
    case .left:  
        print("Hero will block")  
}
```

```
enum Move {  
    case up, down, right, left  
}  
  
let characterNextMove: Move = .right  
  
switch characterNextMove {  
    case Move.up:  
        print("Hero will jump")  
    case Move.down:  
        print("Hero will crouch")  
    case .right:  
        print("Hero will walk")  
    case .left:  
        print("Hero will block")  
}
```

Hero will walk

```
enum Move {  
    case up, down, right, left  
}  
  
let characterNextMove: Move = .right
```

```
switch characterNextMove {  
    case Move.up:  
        print("Hero will jump")  
    case Move.down:  
        print("Hero will crouch")  
    case .right:  
        print("Hero will walk")  
    case .left:  
        print("Hero will block")  
}
```

Hero will walk

Stock Markets



```
enum Market: String {  
    case saoPaulo =  
    "SA"  
  
    case frankfurt = "F"  
  
    case taiwan = "TW"  
  
    case bombay = "BO"  
}
```

Hands on

