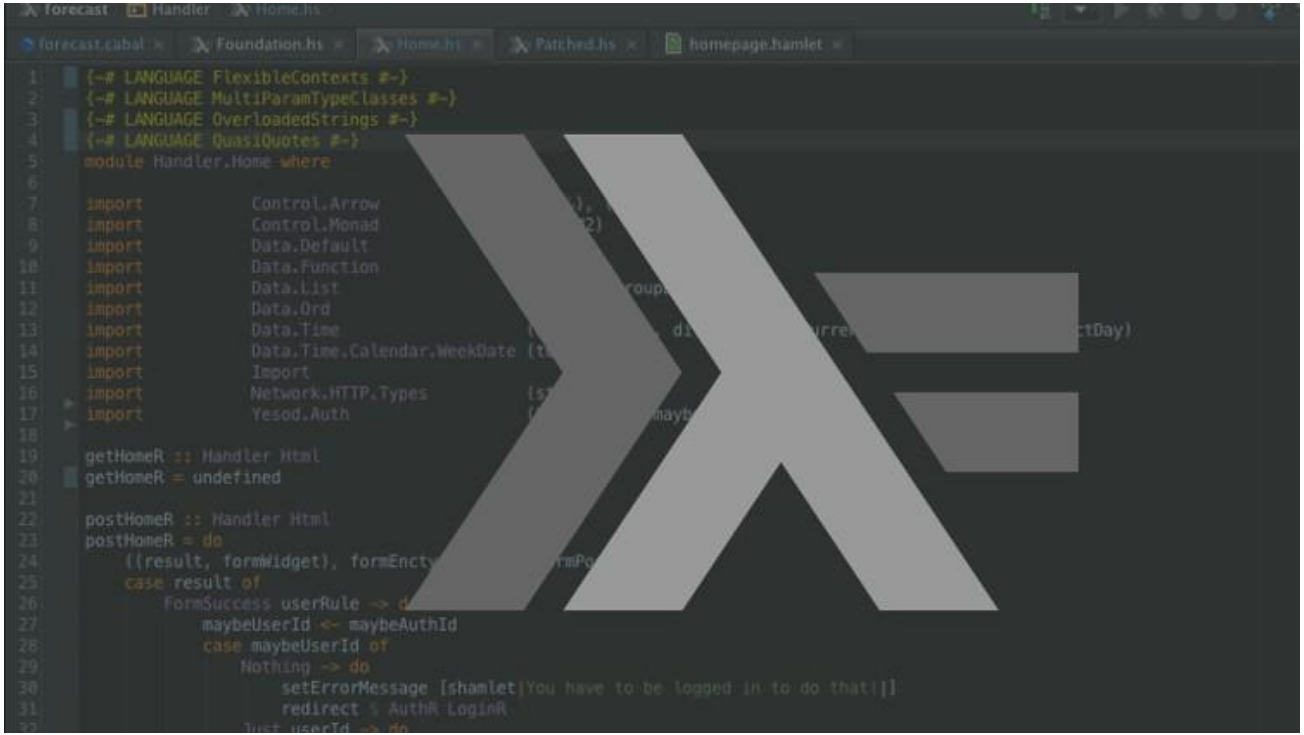
Basic Fundamentals



Expressions

```
λ> 3 + 2 * 2
☞ 7
```

```
λ> (3 + 2) * 2
☞ 10
```

```
λ> even 62
☞ True
```

```
λ> even(62)
☞ True
```

Parentheses **not** necessary.

```
λ> even "Albert"
```

Type error

```
λ> div 14 4
☞ 3
```

Type

```
λ> :type 'D'
☞ 'D' :: Char
```

```
λ> :type "Emma"
☞ "Emma" :: [Char]
```

```
λ> :type not
☞ not :: Bool -> Bool
```

```
λ> :type length
☞ length :: [a] -> Int
```

Factorial

```
factorial :: Integer -> Integer

factorial 0 = 1
factorial n = n * factorial (n - 1)
```

```
λ> factorial 5
☞ 120
```

```
λ> map factorial [0..5]
☞ [1, 1, 2, 6, 24, 120]
```

Summary

- Haskell is pure functional programming language.
- We have made a first taste of its characteristics.
- There are many more things!