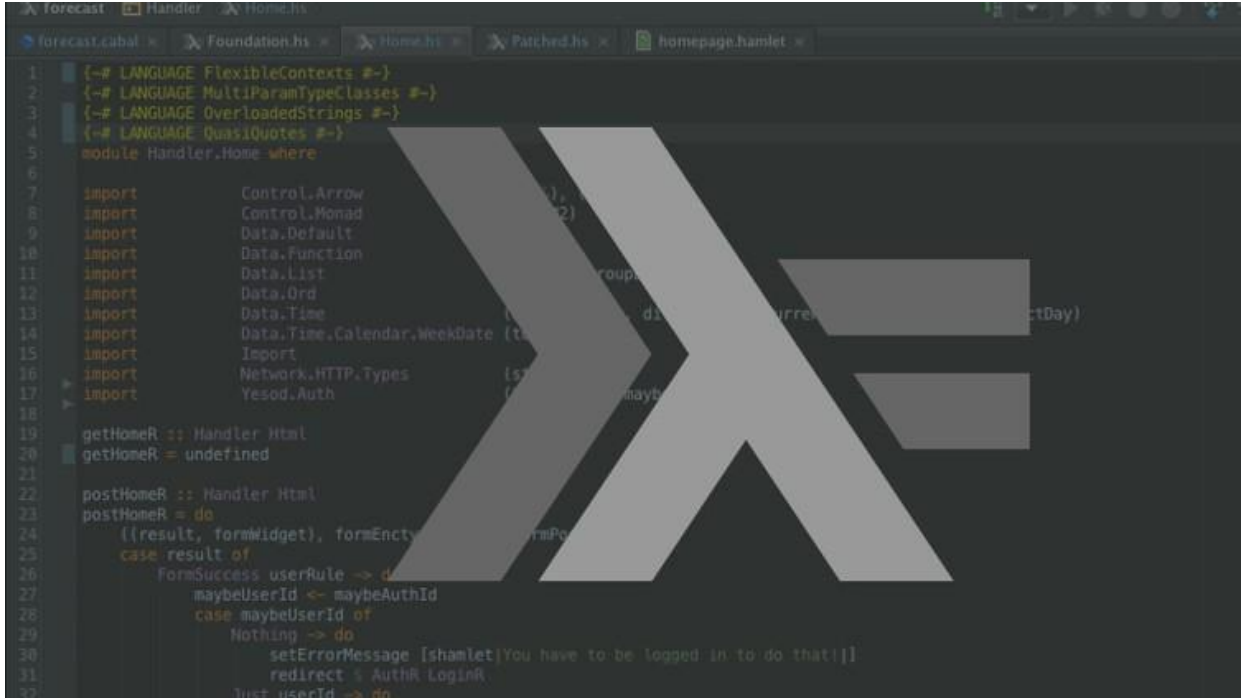
Higher Order Functions



Higher Order Functions

A **Higher Order Function** (HOF) is a function that receives or returns functions

Key point: Functions are first-class objects.

Example in C++:

```
bool compare(int x, int y) {
    return x > y;
}

int main() {
    vector<int> v = { ... };
    sort(v.begin(), v.end(), compare);      // sort is a higher order function
}
```

Higher Order Functions

Example: The predefined function `map` applies a function to each element of a list.

```
map :: (a -> b) -> [a] -> [b]

map f [] = []
map f (x:xs) = f x : map f xs
```

```
λ> map odd [1..5]
☞ [True, False, True, False, True]
```

Higher Order Functions

Example: The predefined function `(.)` returns the composition of two functions:

```
(.) :: (b -> c) -> (a -> b) -> (a -> c)

(f . g) x = f (g x)
```

```
λ> (reverse . sort) [5, 3, 5, 2]
☞ [5, 5, 3, 2]
```

Higher Order Functions

Example: The `apli2` function applies a function to an element twice.

```
apli2 :: (a -> a) -> a -> a

apli2 f x = f (f x)
```

```
λ> apli2 sqrt 16.0
☞ 2.0
```

Equivalently:

```
apli2 :: (a -> a) -> (a -> a)

apli2 f = f . f
```

```
λ> apli2 sqrt 16.0
☞ 2.0
```

Instructor Youtube Channel: Lucas Science

Lucas Science

SUBSCRIBE