

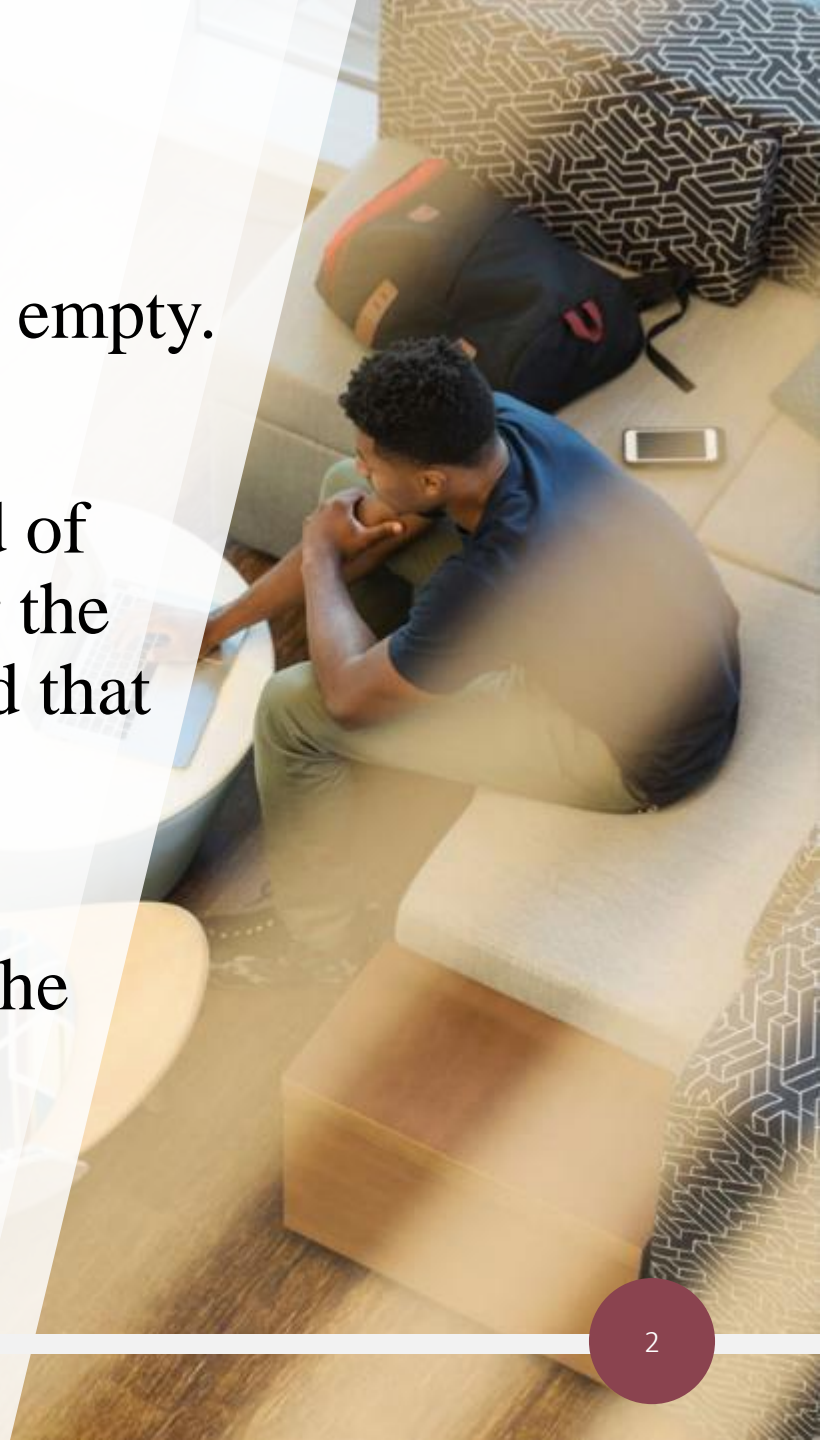
A group of four students are gathered around a table in a library, looking at a laptop screen. The background is filled with bookshelves. The image has a semi-transparent blue overlay on the left side and a semi-transparent red overlay at the bottom.

Streams

Optionals

Optionals

- Think of an *Optional* as a box that may or may not be empty.
- Before Java 8, programmers would return *null* instead of *Optional*. Returning an *Optional* is now a clear way for the API to state that there might not be a value in there (and that the programmer must deal with that).
- In addition to *Optional*<*T*>, there are *Optional*'s for the primitive types:
 - *OptionalDouble*, *OptionalInt* and *OptionalLong*



Optional Examples

```
// a long way to calculate average (just for showing Optional)
public static Optional<Double> calcAverage(int... scores){
    if(scores.length == 0) return Optional.empty();
    int sum=0;
    for(int score:scores) sum += score;
    return Optional.of((double)sum / scores.length);
}
```



Optional Examples

```
Optional<Double> optAvg = calcAverage(50, 60, 70);
// if you do a get() and the Optional is empty you get:
//    NoSuchElementException: No value present
// boolean isPresent() protects us from that.
if(optAvg.isPresent()){
    System.out.println(optAvg.get()); // 60.0
}
// void ifPresent(Consumer c)
optAvg.ifPresent(System.out::println); // 60.0
// T orElse(T t)
System.out.println(optAvg.orElse(Double.NaN)); // 60.0

Optional<Double> optAvg2 = calcAverage(); // will return an empty Optional
System.out.println(optAvg2.orElse(Double.NaN)); // NaN
// T orElseGet(Supplier<T> s)
System.out.println(optAvg2.orElseGet(() -> Math.random())); // 0.8524556508038182
```

Optional Examples

```
public static void doOptionalNull() {  
    Optional<String> optSK = howToDealWithNull("SK");  
    optSK.ifPresent(System.out::println); // SK  
    Optional<String> optNull = howToDealWithNull(null);  
    System.out.println(  
        optNull.orElseGet(  
            () -> "Empty optional")); // Empty optional  
}  
  
public static Optional<String> howToDealWithNull(String param) {  
    // Optional optReturn = param == null ? Optional.empty() : Optional.of(param);  
    Optional optReturn = Optional.ofNullable(param); // same as previous line  
    return optReturn;  
}
```

Optional Examples

```
public static void doOptionalPrimitiveAverage() {  
    OptionalDouble optAvg = IntStream.rangeClosed(1, 10)  
                                     .average();  
    // DoubleConsumer - functional interface; functional method is:  
    //     void accept(double value)  
    optAvg.ifPresent((d) -> System.out.println(d)); // 5.5  
    System.out.println(optAvg.getAsDouble()); // 5.5  
    // DoubleSupplier - functional interface; functional method is:  
    //     double getAsDouble()  
    System.out.println(optAvg.orElseGet(() -> Double.NaN)); // 5.5  
}
```