The main() Function

Learn about the special main function as the entry point into your program.

WE'LL COVER THE FOLLOWING

- Anatomy of a `main` Function
- Quiz
- Exercises
- Summary

Every application requires an entry point where program execution is started. In most languages, including Kotlin, this is called the `main` function.

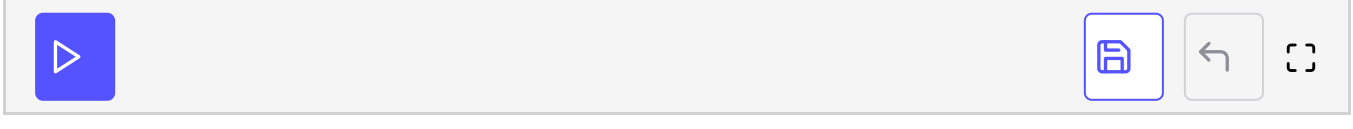
> Kotlin also allows writing *script files* (`.kts` file ending) which don't require an explicit `main` function, similar to Bash or Python scripts. When talking about an "application" above, such scripts are excluded.

In fact, even the code listings here on Educative require a `main` function, although you have not seen one yet. Up until this point, a `main` function was always hidden in code surrounding the code you saw in the listings. This is what enabled the code examples to run.

Anatomy of a `main` Function

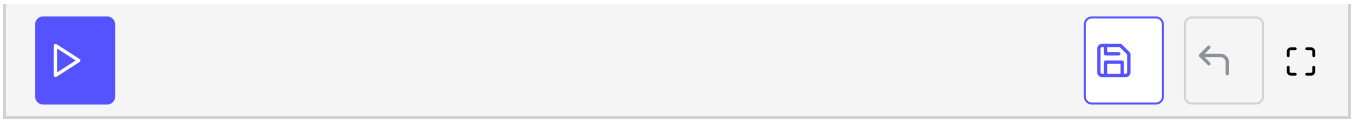
The `main` function in Kotlin is simple:

```
fun main() {
  // Your code here...
}
```



It has the simplest possible signature: no inputs and no outputs*. This has been the case since Kotlin 1.3. Before that, an input parameter was required for command-line arguments:

```
fun main(args: Array<String>) {
  // Your code here...
}
```



Oftentimes, we don't need the command-line arguments in our code. In these cases, we can use the simpler signature.

> *To be more specific, "no output" in Kotlin means that it returns `Unit`. However, this makes no difference for us at this point.

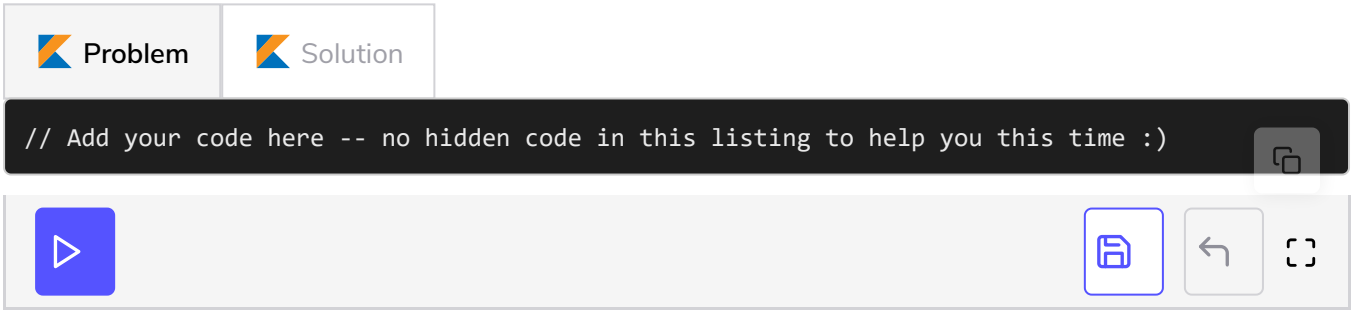
Quiz

The `main` Function

1 Which types of programs requires a `main` function as entry point?

Exercises

Populate the following code listing with your GCD implementation and a `main` function. In your `main` function, print the `gcd(93, 27)`.



Hint: You can call your GCD function from within the `main` function.

Summary

Every Kotlin application you will write requires an entry point in the form of a `main` function.

- You can use the simple signature `fun main()` in most cases.
- To have access to any command-line with which your application was started (e.g., for configuration), you can use `fun main(args: Array<String>)`.

Next up, you'll learn about Kotlin's shorthand notation for functions which can be a great tool to write concise but readable code.