

Built-in classifier types

- TODO: Move the whole section to type system?
- TODO: Move `kotlin.Unit` here?
- TODO: `Appendable/StringBuilder`? depends on how we plan to approach the interpolation expansion
- TODO: `{Builtin}Array` types?

As well as the types defined in the [type system section][Built-in types], Kotlin defines several built-in classifier types that are important for the rest of this document. These have their own declarations in the standard library, but have special semantics in Kotlin.

Note: this is not meant to declare all the types available in the standard library, for this please refer to the standard library documentation .

(TODO: link?)

`kotlin.Boolean`

`kotlin.Boolean` is the boolean logic type of Kotlin, representing the value that may be either `true` or `false`. It is the type of [boolean literals][Boolean literals] as well as the type returned or expected by some built-in Kotlin operators. For other traits of this type (such as the classes it inherits from, interfaces it may inherit from and its member functions) please refer to the standard library specification.

Built-in integer types

There are several built-in class types that represent signed integer numbers of different bit size. Kotlin does not have a built-in infinite-length integer number class. Kotlin also does not currently define any built-in unsigned integer number types . The signed integer number types are:

(TODO: Kotlin 1.3 does)

- `kotlin.Int`
- `kotlin.Short`
- `kotlin.Byte`
- `kotlin.Long`

These types may or may not have different runtime representation. See your platform reference for details.

`kotlin.Int` is the type of integer numbers that is required to be able to hold at least the values in the range from -2^{31} to $2^{31} - 1$. If an arithmetic operation on `kotlin.Int` results in arithmetic overflow or underflow, the result is undefined.

`kotlin.Short` is the type of integer numbers that is required to be able to hold at least the values in the range from -2^{15} to $2^{15} - 1$. If an arithmetic operation on `kotlin.Short` results in arithmetic overflow or underflow, the result is undefined.

`kotlin.Byte` is the type of integer numbers that is required to be able to hold at least the values in the range from -2^7 to $2^7 - 1$. If an arithmetic operation on `kotlin.Byte` results in arithmetic overflow or underflow, the result is undefined.

`kotlin.Long` is the type of integer numbers that is required to be able to hold at least the values in the range from -2^{63} to $2^{63} - 1$. If an arithmetic operation on `kotlin.Long` results in arithmetic overflow or underflow, the result is undefined.

For other traits of these types (such as the classes they inherit from, interfaces they may inherit from and their member functions) please refer to the standard library specification.

Built-in floating point arithmetic types

There are two built-in class types that represent floating-point numbers: `kotlin.Float` and `kotlin.Double`. These types may or may not have different runtime representations. See your platform reference for details.

`kotlin.Float` is the type of floating-point number that is able to contain all the numbers as a IEEE754 single-precision binary floating number with the same precision. `kotlin.Double` is the type of floating-point number that is able to contain all the numbers as a IEEE754 double-precision binary floating number with the same precision.

TODO: or do they?

Platform specification may give a more thorough information on how these types are represented on a particular platform. For other traits of these types (such as the classes they inherit from, interfaces they may inherit from and their member functions) please refer to the standard library specification.

TODO: FP semantics are pretty hard, how much of that we want to put here?

`kotlin.Char`

`kotlin.Char` is the built-in class type that represents a single unicode symbol

in UTF-16 [character encoding](#). It is the type of [\[character literals\]](#)[\[Character literals\]](#). For other traits of this type (such as the classes it inherits from, interfaces it may inherit from and its member functions) please refer to the standard library specification. [\(TODO: link\)](#)

TODO: UTF-16 or UCS-2?

kotlin.String

`kotlin.String` is the built-in class type that represents a sequence of unicode symbol in UTF-16 [character encoding](#). It is the type of the result of [\[string interpolation\]](#)[\[String interpolation expressions\]](#). For other traits of this type (such as the classes it inherits from, interfaces it may inherit from and its member functions) please refer to the standard library specification. [\(TODO: link\)](#)

TODO: UTF-16 or UCS-2?