**Класс String:**

fun compareTo(**other**: String): Int

Compares this object with the specified object for order. Returns zero if this object is equal to the specified [other](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-comparable/compare-to.html#kotlin.Comparable$compareTo(kotlin.Comparable.T)/other) object, a negative number if it's less than [other](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-comparable/compare-to.html#kotlin.Comparable$compareTo(kotlin.Comparable.T)/other), or a positive number if it's greater than [other](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-comparable/compare-to.html#kotlin.Comparable$compareTo(kotlin.Comparable.T)/other).

fun equals(**other**: Any?): Boolean

Indicates whether some other object is "equal to" this one. Implementations must fulfil the following requirements:

fun get(**index**: Int): Char

Returns the character of this string at the specified [index](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-string/get.html#kotlin.String$get(kotlin.Int)/index).

If the [index](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-string/get.html#kotlin.String$get(kotlin.Int)/index) is out of bounds of this string, throws an [IndexOutOfBoundsException](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-index-out-of-bounds-exception/index.html" \l "kotlin.IndexOutOfBoundsException) except in Kotlin/JS where the behavior is unspecified.

fun hashCode(): Int

Returns a hash code value for the object.

operator fun plus(**other**: Any?): String

Returns a string obtained by concatenating this string with the string representation of the given [other](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-string/plus.html#kotlin.String$plus(kotlin.Any?)/other) object.

fun subSequence(**startIndex**: Int, **endIndex**: Int): [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)

Returns a new character sequence that is a subsequence of this character sequence, starting at the specified [startIndex](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/sub-sequence.html" \l "kotlin.CharSequence$subSequence(kotlin.Int,%20kotlin.Int)/startIndex) and ending right before the specified [endIndex](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/sub-sequence.html" \l "kotlin.CharSequence$subSequence(kotlin.Int,%20kotlin.Int)/endIndex).

fun toString(): String

Returns a string representation of the object.

val [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).lastIndex: Int

Returns the index of the last character in the char sequence or -1 if it is empty.

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).all(  
    **predicate**: (Char) -> Boolean  
): Boolean

Returns true if all characters match the given [predicate](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/all.html#kotlin.text$all(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.Boolean)))/predicate).

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).any(): Boolean

Returns true if char sequence has at least one character.

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).any(  
    **predicate**: (Char) -> Boolean  
): Boolean

Returns true if at least one character matches the given [predicate](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/any.html#kotlin.text$any(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.Boolean)))/predicate).

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).asSequence(): [Sequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.sequences/-sequence/index.html)<Char>

Creates a [Sequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.sequences/-sequence/index.html) instance that wraps the original char sequence returning its characters when being iterated.

inline fun <K, V> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).associate(  
    **transform**: (Char) -> [Pair](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-pair/index.html)<K, V>  
): [Map](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-map/index.html)<K, V>

Returns a [Map](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-map/index.html#kotlin.collections.Map) containing key-value pairs provided by [transform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/associate.html#kotlin.text$associate(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.Pair((kotlin.text.associate.K,%20kotlin.text.associate.V)))))/transform) function applied to characters of the given char sequence.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).chunked(**size**: Int): [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<String>

Splits this char sequence into a list of strings each not exceeding the given [size](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/chunked.html#kotlin.text$chunked(kotlin.CharSequence,%20kotlin.Int)/size).

fun <R> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).chunked(  
    **size**: Int,  
    **transform**: ([CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)) -> R  
): [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<R>

Splits this char sequence into several char sequences each not exceeding the given [size](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/chunked.html#kotlin.text$chunked(kotlin.CharSequence,%20kotlin.Int,%20kotlin.Function1((kotlin.CharSequence,%20kotlin.text.chunked.R)))/size) and applies the given [transform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/chunked.html#kotlin.text$chunked(kotlin.CharSequence,%20kotlin.Int,%20kotlin.Function1((kotlin.CharSequence,%20kotlin.text.chunked.R)))/transform) function to an each.

fun String.codePointAt(**index**: Int): Int

Returns the character (Unicode code point) at the specified index.

operator fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).contains(  
    **other**: [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html),  
    **ignoreCase**: Boolean = false  
): Boolean

Returns true if this char sequence contains the specified [other](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/contains.html#kotlin.text$contains(kotlin.CharSequence,%20kotlin.CharSequence,%20kotlin.Boolean)/other) sequence of characters as a substring.

ignoreCase - true to ignore character case when comparing strings. By default false.

operator fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).contains(**regex**: [Regex](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/-regex/index.html)): Boolean

Returns true if this char sequence contains at least one match of the specified regular expression [regex](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/contains.html#kotlin.text$contains(kotlin.CharSequence,%20kotlin.text.Regex)/regex).

infix fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)?.contentEquals(  
    **other**: [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)?  
): Boolean

Returns true if the contents of this char sequence are equal to the contents of the specified [other](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/content-equals.html#kotlin.text$contentEquals(kotlin.CharSequence?,%20kotlin.CharSequence?)/other), i.e. both char sequences contain the same number of the same characters in the same order.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).count(): Int

Returns the length of this char sequence.

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).count(  
    **predicate**: (Char) -> Boolean  
): Int

Returns the number of characters matching the given [predicate](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/count.html#kotlin.text$count(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.Boolean)))/predicate).

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).drop(**n**: Int): [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)

Returns a subsequence of this char sequence with the first [n](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/drop.html#kotlin.text$drop(kotlin.CharSequence,%20kotlin.Int)/n) characters removed.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).dropLast(**n**: Int): [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)

Returns a subsequence of this char sequence with the last [n](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/drop-last.html#kotlin.text$dropLast(kotlin.CharSequence,%20kotlin.Int)/n) characters removed.

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).dropLastWhile(  
    **predicate**: (Char) -> Boolean  
): [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)

Returns a subsequence of this char sequence containing all characters except last characters that satisfy the given [predicate](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/drop-last-while.html#kotlin.text$dropLastWhile(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.Boolean)))/predicate).

inline fun String.dropWhile(  
    **predicate**: (Char) -> Boolean  
): String

Returns a string containing all characters except first characters that satisfy the given [predicate](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/drop-while.html#kotlin.text$dropWhile(kotlin.String,%20kotlin.Function1((kotlin.Char,%20kotlin.Boolean)))/predicate).

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).elementAtOrElse(  
    **index**: Int,  
    **defaultValue**: (Int) -> Char  
): Char

Returns a character at the given [index](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/element-at-or-else.html#kotlin.text$elementAtOrElse(kotlin.CharSequence,%20kotlin.Int,%20kotlin.Function1((kotlin.Int,%20kotlin.Char)))/index) or the result of calling the [defaultValue](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/element-at-or-else.html" \l "kotlin.text$elementAtOrElse(kotlin.CharSequence,%20kotlin.Int,%20kotlin.Function1((kotlin.Int,%20kotlin.Char)))/defaultValue) function if the [index](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/element-at-or-else.html#kotlin.text$elementAtOrElse(kotlin.CharSequence,%20kotlin.Int,%20kotlin.Function1((kotlin.Int,%20kotlin.Char)))/index) is out of bounds of this char sequence.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).elementAtOrNull(**index**: Int): Char?

Returns a character at the given [index](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/element-at-or-null.html#kotlin.text$elementAtOrNull(kotlin.CharSequence,%20kotlin.Int)/index) or null if the [index](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/element-at-or-null.html#kotlin.text$elementAtOrNull(kotlin.CharSequence,%20kotlin.Int)/index) is out of bounds of this char sequence.

fun String.endsWith(  
    **suffix**: String,  
    **ignoreCase**: Boolean  
): Boolean

Returns true if this string ends with the specified suffix.

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).filter(  
    **predicate**: (Char) -> Boolean  
): [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)

Returns a char sequence containing only those characters from the original char sequence that match the given [predicate](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/filter.html#kotlin.text$filter(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.Boolean)))/predicate).

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).filterIndexed(  
    **predicate**: (**index**: Int, Char) -> Boolean  
): [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)

Returns a char sequence containing only those characters from the original char sequence that match the given [predicate](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/filter-indexed.html#kotlin.text$filterIndexed(kotlin.CharSequence,%20kotlin.Function2((kotlin.Int,%20kotlin.Char,%20kotlin.Boolean)))/predicate).

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).filterNot(  
    **predicate**: (Char) -> Boolean  
): [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)

Returns a char sequence containing only those characters from the original char sequence that do not match the given [predicate](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/filter-not.html#kotlin.text$filterNot(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.Boolean)))/predicate).

inline fun <C : [Appendable](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/-appendable/index.html)> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).filterNotTo(  
    **destination**: C,  
    **predicate**: (Char) -> Boolean  
): C

Appends all characters not matching the given [predicate](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/filter-not-to.html#kotlin.text$filterNotTo(kotlin.CharSequence,%20kotlin.text.filterNotTo.C,%20kotlin.Function1((kotlin.Char,%20kotlin.Boolean)))/predicate) to the given [destination](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/filter-not-to.html#kotlin.text$filterNotTo(kotlin.CharSequence,%20kotlin.text.filterNotTo.C,%20kotlin.Function1((kotlin.Char,%20kotlin.Boolean)))/destination).

inline fun <C : [Appendable](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/-appendable/index.html)> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).filterTo(  
    **destination**: C,  
    **predicate**: (Char) -> Boolean  
): C

Appends all characters matching the given [predicate](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/filter-to.html#kotlin.text$filterTo(kotlin.CharSequence,%20kotlin.text.filterTo.C,%20kotlin.Function1((kotlin.Char,%20kotlin.Boolean)))/predicate) to the given [destination](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/filter-to.html#kotlin.text$filterTo(kotlin.CharSequence,%20kotlin.text.filterTo.C,%20kotlin.Function1((kotlin.Char,%20kotlin.Boolean)))/destination).

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).find(  
    **predicate**: (Char) -> Boolean  
): Char?

Returns the first character matching the given [predicate](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/find.html#kotlin.text$find(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.Boolean)))/predicate), or null if no such character was found.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).findAnyOf(  
    **strings**: [Collection](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-collection/index.html)<String>,  
    **startIndex**: Int = 0,  
    **ignoreCase**: Boolean = false  
): [Pair](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-pair/index.html)<Int, String>?

Finds the first occurrence of any of the specified [strings](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/find-any-of.html#kotlin.text$findAnyOf(kotlin.CharSequence,%20kotlin.collections.Collection((kotlin.String)),%20kotlin.Int,%20kotlin.Boolean)/strings) in this char sequence, starting from the specified [startIndex](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/find-any-of.html" \l "kotlin.text$findAnyOf(kotlin.CharSequence,%20kotlin.collections.Collection((kotlin.String)),%20kotlin.Int,%20kotlin.Boolean)/startIndex) and optionally ignoring the case.

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).findLast(  
    **predicate**: (Char) -> Boolean  
): Char?

Returns the last character matching the given [predicate](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/find-last.html#kotlin.text$findLast(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.Boolean)))/predicate), or null if no such character was found.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).findLastAnyOf(  
    **strings**: [Collection](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-collection/index.html)<String>,  
    **startIndex**: Int = lastIndex,  
    **ignoreCase**: Boolean = false  
): [Pair](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-pair/index.html)<Int, String>?

Finds the last occurrence of any of the specified [strings](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/find-last-any-of.html#kotlin.text$findLastAnyOf(kotlin.CharSequence,%20kotlin.collections.Collection((kotlin.String)),%20kotlin.Int,%20kotlin.Boolean)/strings) in this char sequence, starting from the specified [startIndex](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/find-last-any-of.html" \l "kotlin.text$findLastAnyOf(kotlin.CharSequence,%20kotlin.collections.Collection((kotlin.String)),%20kotlin.Int,%20kotlin.Boolean)/startIndex) and optionally ignoring the case.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).first(): Char

Returns first character.

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).first(  
    **predicate**: (Char) -> Boolean  
): Char

Returns the first character matching the given [predicate](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/first.html#kotlin.text$first(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.Boolean)))/predicate).

inline fun <R : Any> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).firstNotNullOf(  
    **transform**: (Char) -> R?  
): R

Returns the first non-null value produced by [transform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/first-not-null-of.html#kotlin.text$firstNotNullOf(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.text.firstNotNullOf.R?)))/transform) function being applied to characters of this char sequence in iteration order, or throws [NoSuchElementException](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-no-such-element-exception/index.html" \l "kotlin.NoSuchElementException) if no non-null value was produced.

inline fun <R : Any> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).firstNotNullOfOrNull(  
    **transform**: (Char) -> R?  
): R?

Returns the first non-null value produced by [transform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/first-not-null-of-or-null.html#kotlin.text$firstNotNullOfOrNull(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.text.firstNotNullOfOrNull.R?)))/transform) function being applied to characters of this char sequence in iteration order, or null if no non-null value was produced.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).firstOrNull(): Char?

Returns the first character, or null if the char sequence is empty.

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).firstOrNull(  
    **predicate**: (Char) -> Boolean  
): Char?

Returns the first character matching the given [predicate](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/first-or-null.html#kotlin.text$firstOrNull(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.Boolean)))/predicate), or null if character was not found.

inline fun <R> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).flatMap(  
    **transform**: (Char) -> [Iterable](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-iterable/index.html)<R>  
): [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<R>

Returns a single list of all elements yielded from results of [transform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/flat-map.html#kotlin.text$flatMap(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.collections.Iterable((kotlin.text.flatMap.R)))))/transform) function being invoked on each character of original char sequence.

inline fun <R> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).flatMapIndexed(  
    **transform**: (**index**: Int, Char) -> [Iterable](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-iterable/index.html)<R>  
): [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<R>

Returns a single list of all elements yielded from results of [transform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/flat-map-indexed.html#kotlin.text$flatMapIndexed(kotlin.CharSequence,%20kotlin.Function2((kotlin.Int,%20kotlin.Char,%20kotlin.collections.Iterable((kotlin.text.flatMapIndexed.R)))))/transform) function being invoked on each character and its index in the original char sequence.

inline fun <R, C : [MutableCollection](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-mutable-collection/index.html)<in R>> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).flatMapIndexedTo(  
    **destination**: C,  
    **transform**: (**index**: Int, Char) -> [Iterable](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-iterable/index.html)<R>  
): C

Appends all elements yielded from results of [transform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/flat-map-indexed-to.html#kotlin.text$flatMapIndexedTo(kotlin.CharSequence,%20kotlin.text.flatMapIndexedTo.C,%20kotlin.Function2((kotlin.Int,%20kotlin.Char,%20kotlin.collections.Iterable((kotlin.text.flatMapIndexedTo.R)))))/transform) function being invoked on each character and its index in the original char sequence, to the given [destination](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/flat-map-indexed-to.html#kotlin.text$flatMapIndexedTo(kotlin.CharSequence,%20kotlin.text.flatMapIndexedTo.C,%20kotlin.Function2((kotlin.Int,%20kotlin.Char,%20kotlin.collections.Iterable((kotlin.text.flatMapIndexedTo.R)))))/destination).

inline fun <R, C : [MutableCollection](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-mutable-collection/index.html)<in R>> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).flatMapTo(  
    **destination**: C,  
    **transform**: (Char) -> [Iterable](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-iterable/index.html)<R>  
): C

Appends all elements yielded from results of [transform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/flat-map-to.html#kotlin.text$flatMapTo(kotlin.CharSequence,%20kotlin.text.flatMapTo.C,%20kotlin.Function1((kotlin.Char,%20kotlin.collections.Iterable((kotlin.text.flatMapTo.R)))))/transform) function being invoked on each character of original char sequence, to the given [destination](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/flat-map-to.html#kotlin.text$flatMapTo(kotlin.CharSequence,%20kotlin.text.flatMapTo.C,%20kotlin.Function1((kotlin.Char,%20kotlin.collections.Iterable((kotlin.text.flatMapTo.R)))))/destination).

inline fun <R> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).fold(  
    **initial**: R,  
    **operation**: (**acc**: R, Char) -> R  
): R

Accumulates value starting with [initial](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/fold.html#kotlin.text$fold(kotlin.CharSequence,%20kotlin.text.fold.R,%20kotlin.Function2((kotlin.text.fold.R,%20kotlin.Char,%20)))/initial) value and applying [operation](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/fold.html#kotlin.text$fold(kotlin.CharSequence,%20kotlin.text.fold.R,%20kotlin.Function2((kotlin.text.fold.R,%20kotlin.Char,%20)))/operation) from left to right to current accumulator value and each character.

Returns the specified [initial](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/fold.html#kotlin.text$fold(kotlin.CharSequence,%20kotlin.text.fold.R,%20kotlin.Function2((kotlin.text.fold.R,%20kotlin.Char,%20)))/initial) value if the char sequence is empty.

operation - function that takes current accumulator value and a character, and calculates the next accumulator value.

inline fun <R> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).foldIndexed(  
    **initial**: R,  
    **operation**: (**index**: Int, **acc**: R, Char) -> R  
): R

Accumulates value starting with [initial](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/fold-indexed.html#kotlin.text$foldIndexed(kotlin.CharSequence,%20kotlin.text.foldIndexed.R,%20kotlin.Function3((kotlin.Int,%20kotlin.text.foldIndexed.R,%20kotlin.Char,%20)))/initial) value and applying [operation](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/fold-indexed.html#kotlin.text$foldIndexed(kotlin.CharSequence,%20kotlin.text.foldIndexed.R,%20kotlin.Function3((kotlin.Int,%20kotlin.text.foldIndexed.R,%20kotlin.Char,%20)))/operation) from left to right to current accumulator value and each character with its index in the original char sequence.

inline fun <R> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).foldRight(  
    **initial**: R,  
    **operation**: (Char, **acc**: R) -> R  
): R

Accumulates value starting with [initial](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/fold-right.html#kotlin.text$foldRight(kotlin.CharSequence,%20kotlin.text.foldRight.R,%20kotlin.Function2((kotlin.Char,%20kotlin.text.foldRight.R,%20)))/initial) value and applying [operation](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/fold-right.html#kotlin.text$foldRight(kotlin.CharSequence,%20kotlin.text.foldRight.R,%20kotlin.Function2((kotlin.Char,%20kotlin.text.foldRight.R,%20)))/operation) from right to left to each character and current accumulator value.

inline fun <R> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).foldRightIndexed(  
    **initial**: R,  
    **operation**: (**index**: Int, Char, **acc**: R) -> R  
): R

Accumulates value starting with [initial](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/fold-right-indexed.html#kotlin.text$foldRightIndexed(kotlin.CharSequence,%20kotlin.text.foldRightIndexed.R,%20kotlin.Function3((kotlin.Int,%20kotlin.Char,%20kotlin.text.foldRightIndexed.R,%20)))/initial) value and applying [operation](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/fold-right-indexed.html#kotlin.text$foldRightIndexed(kotlin.CharSequence,%20kotlin.text.foldRightIndexed.R,%20kotlin.Function3((kotlin.Int,%20kotlin.Char,%20kotlin.text.foldRightIndexed.R,%20)))/operation) from right to left to each character with its index in the original char sequence and current accumulator value.

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).forEach(**action**: (Char) -> Unit)

Performs the given [action](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/for-each.html#kotlin.text$forEach(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.Unit)))/action) on each character.

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).forEachIndexed(  
    **action**: (**index**: Int, Char) -> Unit)

Performs the given [action](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/for-each-indexed.html#kotlin.text$forEachIndexed(kotlin.CharSequence,%20kotlin.Function2((kotlin.Int,%20kotlin.Char,%20kotlin.Unit)))/action) on each character, providing sequential index with the character.

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).getOrElse(  
    **index**: Int,  
    **defaultValue**: (Int) -> Char  
): Char

Returns a character at the given [index](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/get-or-else.html#kotlin.text$getOrElse(kotlin.CharSequence,%20kotlin.Int,%20kotlin.Function1((kotlin.Int,%20kotlin.Char)))/index) or the result of calling the [defaultValue](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/get-or-else.html" \l "kotlin.text$getOrElse(kotlin.CharSequence,%20kotlin.Int,%20kotlin.Function1((kotlin.Int,%20kotlin.Char)))/defaultValue) function if the [index](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/get-or-else.html#kotlin.text$getOrElse(kotlin.CharSequence,%20kotlin.Int,%20kotlin.Function1((kotlin.Int,%20kotlin.Char)))/index) is out of bounds of this char sequence.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).getOrNull(**index**: Int): Char?

Returns a character at the given [index](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/get-or-null.html#kotlin.text$getOrNull(kotlin.CharSequence,%20kotlin.Int)/index) or null if the [index](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/get-or-null.html#kotlin.text$getOrNull(kotlin.CharSequence,%20kotlin.Int)/index) is out of bounds of this char sequence.

inline fun <K> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).groupBy(  
    **keySelector**: (Char) -> K  
): [Map](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-map/index.html)<K, [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<Char>>

Groups characters of the original char sequence by the key returned by the given [keySelector](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/group-by.html" \l "kotlin.text$groupBy(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.text.groupBy.K)))/keySelector) function applied to each character and returns a map where each group key is associated with a list of corresponding characters.

The returned map preserves the entry iteration order of the keys produced from the original char sequence.

inline fun <K, V> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).groupBy(  
    **keySelector**: (Char) -> K,  
    **valueTransform**: (Char) -> V  
): [Map](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-map/index.html)<K, [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<V>>

Groups values returned by the [valueTransform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/group-by.html" \l "kotlin.text$groupBy(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.text.groupBy.K)),%20kotlin.Function1((kotlin.Char,%20kotlin.text.groupBy.V)))/valueTransform) function applied to each character of the original char sequence by the key returned by the given [keySelector](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/group-by.html" \l "kotlin.text$groupBy(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.text.groupBy.K)),%20kotlin.Function1((kotlin.Char,%20kotlin.text.groupBy.V)))/keySelector) function applied to the character and returns a map where each group key is associated with a list of corresponding values.

inline fun <K, M : [MutableMap](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-mutable-map/index.html)<in K, [MutableList](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-mutable-list/index.html)<Char>>> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).groupByTo(  
    **destination**: M,  
    **keySelector**: (Char) -> K  
): M

Groups characters of the original char sequence by the key returned by the given [keySelector](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/group-by-to.html" \l "kotlin.text$groupByTo(kotlin.CharSequence,%20kotlin.text.groupByTo.M,%20kotlin.Function1((kotlin.Char,%20kotlin.text.groupByTo.K)))/keySelector) function applied to each character and puts to the [destination](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/group-by-to.html#kotlin.text$groupByTo(kotlin.CharSequence,%20kotlin.text.groupByTo.M,%20kotlin.Function1((kotlin.Char,%20kotlin.text.groupByTo.K)))/destination) map each group key associated with a list of corresponding characters.

inline fun <K, V, M : [MutableMap](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-mutable-map/index.html)<in K, [MutableList](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-mutable-list/index.html)<V>>> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).groupByTo(  
    **destination**: M,  
    **keySelector**: (Char) -> K,  
    **valueTransform**: (Char) -> V  
): M

Groups values returned by the [valueTransform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/group-by-to.html" \l "kotlin.text$groupByTo(kotlin.CharSequence,%20kotlin.text.groupByTo.M,%20kotlin.Function1((kotlin.Char,%20kotlin.text.groupByTo.K)),%20kotlin.Function1((kotlin.Char,%20kotlin.text.groupByTo.V)))/valueTransform) function applied to each character of the original char sequence by the key returned by the given [keySelector](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/group-by-to.html" \l "kotlin.text$groupByTo(kotlin.CharSequence,%20kotlin.text.groupByTo.M,%20kotlin.Function1((kotlin.Char,%20kotlin.text.groupByTo.K)),%20kotlin.Function1((kotlin.Char,%20kotlin.text.groupByTo.V)))/keySelector) function applied to the character and puts to the [destination](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/group-by-to.html#kotlin.text$groupByTo(kotlin.CharSequence,%20kotlin.text.groupByTo.M,%20kotlin.Function1((kotlin.Char,%20kotlin.text.groupByTo.K)),%20kotlin.Function1((kotlin.Char,%20kotlin.text.groupByTo.V)))/destination) map each group key associated with a list of corresponding values.

inline fun <K> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).groupingBy(  
    crossinline **keySelector**: (Char) -> K  
): [Grouping](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-grouping/index.html)<Char, K>

Creates a [Grouping](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-grouping/index.html) source from a char sequence to be used later with one of group-and-fold operations using the specified [keySelector](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/grouping-by.html" \l "kotlin.text$groupingBy(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.text.groupingBy.K)))/keySelector) function to extract a key from each character.

inline fun <C, R> C.ifBlank(  
    **defaultValue**: () -> R  
): R where C : [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html), C : R

Returns this char sequence if it is not empty and doesn't consist solely of whitespace characters, or the result of calling [defaultValue](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/if-blank.html" \l "kotlin.text$ifBlank(kotlin.text.ifBlank.C,%20kotlin.Function0((kotlin.text.ifBlank.R)))/defaultValue) function otherwise.

inline fun <C, R> C.ifEmpty(  
    **defaultValue**: () -> R  
): R where C : [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html), C : R

Returns this char sequence if it's not empty or the result of calling [defaultValue](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/if-empty.html" \l "kotlin.text$ifEmpty(kotlin.text.ifEmpty.C,%20kotlin.Function0((kotlin.text.ifEmpty.R)))/defaultValue) function if the char sequence is empty.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).indexOf(  
    **char**: Char,  
    **startIndex**: Int = 0,  
    **ignoreCase**: Boolean = false  
): Int

Returns the index within this string of the first occurrence of the specified character, starting from the specified [startIndex](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/index-of.html" \l "kotlin.text$indexOf(kotlin.CharSequence,%20kotlin.Char,%20kotlin.Int,%20kotlin.Boolean)/startIndex). **(+modifications)**

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).isEmpty(): Boolean

Returns true if this char sequence is empty (contains no characters).

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).isNotBlank(): Boolean

Returns true if this char sequence is not empty and contains some characters except of whitespace characters.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).isNotEmpty(): Boolean

Returns true if this char sequence is not empty.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)?.isNullOrBlank(): Boolean

Returns true if this nullable char sequence is either null or empty or consists solely of whitespace characters.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)?.isNullOrEmpty(): Boolean

Returns true if this nullable char sequence is either null or empty.

operator fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).iterator(): [CharIterator](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-char-iterator/index.html)

Iterator for characters of the given char sequence.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).last(): Char

Returns the last character.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).lastIndexOf(  
    **char**: Char,  
    **startIndex**: Int = lastIndex,  
    **ignoreCase**: Boolean = false  
): Int

Returns the index within this char sequence of the last occurrence of the specified character, starting from the specified [startIndex](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/last-index-of.html" \l "kotlin.text$lastIndexOf(kotlin.CharSequence,%20kotlin.Char,%20kotlin.Int,%20kotlin.Boolean)/startIndex).

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).lastIndexOfAny(  
    **chars**: [CharArray](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-array/index.html),  
    **startIndex**: Int = lastIndex,  
    **ignoreCase**: Boolean = false  
): Int

Finds the index of the last occurrence of any of the specified [chars](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/last-index-of-any.html#kotlin.text$lastIndexOfAny(kotlin.CharSequence,%20kotlin.CharArray,%20kotlin.Int,%20kotlin.Boolean)/chars) in this char sequence, starting from the specified [startIndex](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/last-index-of-any.html" \l "kotlin.text$lastIndexOfAny(kotlin.CharSequence,%20kotlin.CharArray,%20kotlin.Int,%20kotlin.Boolean)/startIndex) and optionally ignoring the case.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).lastOrNull(): Char?

Returns the last character, or null if the char sequence is empty.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).lines(): [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<String>

Splits this char sequence to a list of lines delimited by any of the following character sequences: CRLF, LF or CR.

The lines returned do not include terminating line separators.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).lineSequence(): [Sequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.sequences/-sequence/index.html)<String>

Splits this char sequence to a sequence of lines delimited by any of the following character sequences: CRLF, LF or CR.

fun Char.lowercase(): String

Converts this character to lower case using Unicode mapping rules of the invariant locale.

inline fun <R> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).map(  
    **transform**: (Char) -> R  
): [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<R>

Returns a list containing the results of applying the given [transform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/map.html#kotlin.text$map(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.text.map.R)))/transform) function to each character in the original char sequence.

inline fun <R> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).mapIndexed(  
    **transform**: (**index**: Int, Char) -> R  
): [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<R>

Returns a list containing the results of applying the given [transform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/map-indexed.html#kotlin.text$mapIndexed(kotlin.CharSequence,%20kotlin.Function2((kotlin.Int,%20kotlin.Char,%20kotlin.text.mapIndexed.R)))/transform) function to each character and its index in the original char sequence.

transform - function that takes the index of a character and the character itself and returns the result of the transform applied to the character.

inline fun <R : Any> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).mapIndexedNotNull(  
    **transform**: (**index**: Int, Char) -> R?  
): [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<R>

Returns a list containing only the non-null results of applying the given [transform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/map-indexed-not-null.html#kotlin.text$mapIndexedNotNull(kotlin.CharSequence,%20kotlin.Function2((kotlin.Int,%20kotlin.Char,%20kotlin.text.mapIndexedNotNull.R?)))/transform) function to each character and its index in the original char sequence.

inline fun <R : Any, C : [MutableCollection](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-mutable-collection/index.html)<in R>> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).mapIndexedNotNullTo(  
    **destination**: C,  
    **transform**: (**index**: Int, Char) -> R?  
): C

Applies the given [transform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/map-indexed-not-null-to.html#kotlin.text$mapIndexedNotNullTo(kotlin.CharSequence,%20kotlin.text.mapIndexedNotNullTo.C,%20kotlin.Function2((kotlin.Int,%20kotlin.Char,%20kotlin.text.mapIndexedNotNullTo.R?)))/transform) function to each character and its index in the original char sequence and appends only the non-null results to the given [destination](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/map-indexed-not-null-to.html#kotlin.text$mapIndexedNotNullTo(kotlin.CharSequence,%20kotlin.text.mapIndexedNotNullTo.C,%20kotlin.Function2((kotlin.Int,%20kotlin.Char,%20kotlin.text.mapIndexedNotNullTo.R?)))/destination).

inline fun <R, C : [MutableCollection](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-mutable-collection/index.html)<in R>> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).mapIndexedTo(  
    **destination**: C,  
    **transform**: (**index**: Int, Char) -> R  
): C

Applies the given [transform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/map-indexed-to.html#kotlin.text$mapIndexedTo(kotlin.CharSequence,%20kotlin.text.mapIndexedTo.C,%20kotlin.Function2((kotlin.Int,%20kotlin.Char,%20kotlin.text.mapIndexedTo.R)))/transform) function to each character and its index in the original char sequence and appends the results to the given [destination](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/map-indexed-to.html#kotlin.text$mapIndexedTo(kotlin.CharSequence,%20kotlin.text.mapIndexedTo.C,%20kotlin.Function2((kotlin.Int,%20kotlin.Char,%20kotlin.text.mapIndexedTo.R)))/destination).

inline fun <R : Any> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).mapNotNull(  
    **transform**: (Char) -> R?  
): [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<R>

Returns a list containing only the non-null results of applying the given [transform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/map-not-null.html#kotlin.text$mapNotNull(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.text.mapNotNull.R?)))/transform) function to each character in the original char sequence.

inline fun <R : Any, C : [MutableCollection](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-mutable-collection/index.html)<in R>> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).mapNotNullTo(  
    **destination**: C,  
    **transform**: (Char) -> R?  
): C

Applies the given [transform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/map-not-null-to.html#kotlin.text$mapNotNullTo(kotlin.CharSequence,%20kotlin.text.mapNotNullTo.C,%20kotlin.Function1((kotlin.Char,%20kotlin.text.mapNotNullTo.R?)))/transform) function to each character in the original char sequence and appends only the non-null results to the given [destination](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/map-not-null-to.html#kotlin.text$mapNotNullTo(kotlin.CharSequence,%20kotlin.text.mapNotNullTo.C,%20kotlin.Function1((kotlin.Char,%20kotlin.text.mapNotNullTo.R?)))/destination).

inline fun <R, C : [MutableCollection](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-mutable-collection/index.html)<in R>> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).mapTo(  
    **destination**: C,  
    **transform**: (Char) -> R  
): C

Applies the given [transform](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/map-to.html#kotlin.text$mapTo(kotlin.CharSequence,%20kotlin.text.mapTo.C,%20kotlin.Function1((kotlin.Char,%20kotlin.text.mapTo.R)))/transform) function to each character of the original char sequence and appends the results to the given [destination](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/map-to.html#kotlin.text$mapTo(kotlin.CharSequence,%20kotlin.text.mapTo.C,%20kotlin.Function1((kotlin.Char,%20kotlin.text.mapTo.R)))/destination).

infix fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).matches(**regex**: [Regex](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/-regex/index.html)): Boolean

Returns true if this char sequence matches the given regular expression.

inline fun <R : [Comparable](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-comparable/index.html)<R>> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).maxByOrNull(  
    **selector**: (Char) -> R  
): Char?

Returns the first character yielding the largest value of the given function or null if there are no characters.

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).maxOf(  
    **selector**: (Char) -> Double  
): Double

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).maxOf(  
    **selector**: (Char) -> Float  
): Float

Returns the largest value among all values produced by [selector](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/max-of.html#kotlin.text$maxOf(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.Double)))/selector) function applied to each character in the char sequence.

If any of values produced by [selector](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/max-of.html#kotlin.text$maxOf(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.Double)))/selector) function is NaN, the returned result is NaN.

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).maxOfOrNull(  
    **selector**: (Char) -> Double  
): Double?

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).maxOfOrNull(  
    **selector**: (Char) -> Float  
): Float?

Returns the largest value among all values produced by [selector](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/max-of-or-null.html#kotlin.text$maxOfOrNull(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.Double)))/selector) function applied to each character in the char sequence or null if there are no characters.

inline fun <R> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).maxOfWith(  
    **comparator**: [Comparator](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-comparator/index.html)<in R>,  
    **selector**: (Char) -> R  
): R

Returns the largest value according to the provided [comparator](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/max-of-with.html#kotlin.text$maxOfWith(kotlin.CharSequence,%20kotlin.Comparator((kotlin.text.maxOfWith.R)),%20kotlin.Function1((kotlin.Char,%20kotlin.text.maxOfWith.R)))/comparator) among all values produced by [selector](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/max-of-with.html#kotlin.text$maxOfWith(kotlin.CharSequence,%20kotlin.Comparator((kotlin.text.maxOfWith.R)),%20kotlin.Function1((kotlin.Char,%20kotlin.text.maxOfWith.R)))/selector) function applied to each character in the char sequence.

inline fun <R> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).maxOfWithOrNull(  
    **comparator**: [Comparator](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-comparator/index.html)<in R>,  
    **selector**: (Char) -> R  
): R?

Returns the largest value according to the provided [comparator](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/max-of-with-or-null.html#kotlin.text$maxOfWithOrNull(kotlin.CharSequence,%20kotlin.Comparator((kotlin.text.maxOfWithOrNull.R)),%20kotlin.Function1((kotlin.Char,%20kotlin.text.maxOfWithOrNull.R)))/comparator) among all values produced by [selector](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/max-of-with-or-null.html#kotlin.text$maxOfWithOrNull(kotlin.CharSequence,%20kotlin.Comparator((kotlin.text.maxOfWithOrNull.R)),%20kotlin.Function1((kotlin.Char,%20kotlin.text.maxOfWithOrNull.R)))/selector) function applied to each character in the char sequence or null if there are no characters.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).maxOrNull(): Char?

Returns the largest character or null if there are no characters.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).maxWithOrNull(  
    **comparator**: [Comparator](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-comparator/index.html)<in Char>  
): Char?

Returns the first character having the largest value according to the provided [comparator](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/max-with-or-null.html#kotlin.text$maxWithOrNull(kotlin.CharSequence,%20kotlin.Comparator((kotlin.Char)))/comparator) or null if there are no characters.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).none(): Boolean

Returns true if the char sequence has no characters.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).padEnd(  
    **length**: Int,  
    **padChar**: Char = ' '  
): [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)

Returns a char sequence with content of this char sequence padded at the end to the specified [length](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/pad-end.html#kotlin.text$padEnd(kotlin.CharSequence,%20kotlin.Int,%20kotlin.Char)/length) with the specified character or space.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).padStart(  
    **length**: Int,  
    **padChar**: Char = ' '  
): [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)

Returns a char sequence with content of this char sequence padded at the beginning to the specified [length](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/pad-start.html#kotlin.text$padStart(kotlin.CharSequence,%20kotlin.Int,%20kotlin.Char)/length) with the specified character or space.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).random(): Char

Returns a random character from this char sequence.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).randomOrNull(): Char?

Returns a random character from this char sequence, or null if this char sequence is empty.

operator fun <T : [Comparable](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-comparable/index.html)<T>> T.rangeTo(  
    **that**: T  
): [ClosedRange](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/-closed-range/index.html)<T>

Creates a range from this [Comparable](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-comparable/index.html#kotlin.Comparable) value to the specified [that](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/range-to.html#kotlin.ranges$rangeTo(kotlin.ranges.rangeTo.T,%20kotlin.ranges.rangeTo.T)/that) value.

This value needs to be smaller than or equal to [that](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/range-to.html#kotlin.ranges$rangeTo(kotlin.ranges.rangeTo.T,%20kotlin.ranges.rangeTo.T)/that) value, otherwise the returned range will be empty.

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).reduce(  
    **operation**: (**acc**: Char, Char) -> Char  
): Char

Accumulates value starting with the first character and applying [operation](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/reduce.html#kotlin.text$reduce(kotlin.CharSequence,%20kotlin.Function2((kotlin.Char,%20,%20)))/operation) from left to right to current accumulator value and each character.

Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way, please use [reduceOrNull](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/reduce-or-null.html) instead. It returns null when its receiver is empty.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).removeRange(  
    **startIndex**: Int,  
    **endIndex**: Int  
): [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)

Returns a char sequence with content of this char sequence where its part at the given range is removed.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).removeSuffix(  
    **suffix**: [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)  
): [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)

If this char sequence ends with the given [suffix](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/remove-suffix.html#kotlin.text$removeSuffix(kotlin.CharSequence,%20kotlin.CharSequence)/suffix), returns a new char sequence with the suffix removed. Otherwise, returns a new char sequence with the same characters.

fun String.replace(  
    **oldChar**: Char,  
    **newChar**: Char,  
    **ignoreCase**: Boolean  
): String

Returns a new string with all occurrences of [oldChar](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/replace.html" \l "kotlin.text$replace(kotlin.String,%20kotlin.Char,%20kotlin.Char,%20kotlin.Boolean)/oldChar) replaced with [newChar](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/replace.html" \l "kotlin.text$replace(kotlin.String,%20kotlin.Char,%20kotlin.Char,%20kotlin.Boolean)/newChar).

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).reversed(): [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html)

Returns a char sequence with characters in reversed order.

inline fun <R> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).runningFold(  
    **initial**: R,  
    **operation**: (**acc**: R, Char) -> R  
): [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<R>

Returns a list containing successive accumulation values generated by applying [operation](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/running-fold.html#kotlin.text$runningFold(kotlin.CharSequence,%20kotlin.text.runningFold.R,%20kotlin.Function2((kotlin.text.runningFold.R,%20kotlin.Char,%20)))/operation) from left to right to each character and current accumulator value that starts with [initial](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/running-fold.html#kotlin.text$runningFold(kotlin.CharSequence,%20kotlin.text.runningFold.R,%20kotlin.Function2((kotlin.text.runningFold.R,%20kotlin.Char,%20)))/initial) value.

Note that acc value passed to [operation](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/running-fold.html#kotlin.text$runningFold(kotlin.CharSequence,%20kotlin.text.runningFold.R,%20kotlin.Function2((kotlin.text.runningFold.R,%20kotlin.Char,%20)))/operation) function should not be mutated; otherwise it would affect the previous value in resulting list.

inline fun <R> [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).scan(  
    **initial**: R,  
    **operation**: (**acc**: R, Char) -> R  
): [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<R>

Returns a list containing successive accumulation values generated by applying [operation](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/scan.html#kotlin.text$scan(kotlin.CharSequence,%20kotlin.text.scan.R,%20kotlin.Function2((kotlin.text.scan.R,%20kotlin.Char,%20)))/operation) from left to right to each character and current accumulator value that starts with [initial](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/scan.html#kotlin.text$scan(kotlin.CharSequence,%20kotlin.text.scan.R,%20kotlin.Function2((kotlin.text.scan.R,%20kotlin.Char,%20)))/initial) value.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).single(): Char

Returns the single character, or throws an exception if the char sequence is empty or has more than one character.

inline fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).single(  
    **predicate**: (Char) -> Boolean  
): Char

Returns the single character matching the given [predicate](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/single.html#kotlin.text$single(kotlin.CharSequence,%20kotlin.Function1((kotlin.Char,%20kotlin.Boolean)))/predicate), or throws exception if there is no or more than one matching character.

fun [CharSequence](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-char-sequence/index.html).split(  
    vararg **delimiters**: String,  
    **ignoreCase**: Boolean = false,  
    **limit**: Int = 0  
): [List](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-list/index.html)<String>

Splits this char sequence to a list of strings around occurrences of the specified [delimiters](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/split.html#kotlin.text$split(kotlin.CharSequence,%20kotlin.Array((kotlin.String)),%20kotlin.Boolean,%20kotlin.Int)/delimiters).

delimiters - One or more strings to be used as delimiters.

ignoreCase - true to ignore character case when matching a delimiter. By default false.

limit - The maximum number of substrings to return. Zero by default means no limit is set.

To avoid ambiguous results when strings in [delimiters](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/split.html#kotlin.text$split(kotlin.CharSequence,%20kotlin.Array((kotlin.String)),%20kotlin.Boolean,%20kotlin.Int)/delimiters) have characters in common, this method proceeds from the beginning to the end of this string, and matches at each position the first element in [delimiters](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/split.html#kotlin.text$split(kotlin.CharSequence,%20kotlin.Array((kotlin.String)),%20kotlin.Boolean,%20kotlin.Int)/delimiters) that is equal to a delimiter in this instance at that position.

**Класс Int:**

infix fun and(**other**: Int): Int

Performs a bitwise AND operation between the two values.

operator fun compareTo(**other**: Byte): Int

operator fun compareTo(**other**: Short): Int

operator fun compareTo(**other**: Int): Int

operator fun compareTo(**other**: Long): Int

operator fun compareTo(**other**: Float): Int

operator fun compareTo(**other**: Double): Int

Compares this value with the specified value for order. Returns zero if this value is equal to the specified other value, a negative number if it's less than other, or a positive number if it's greater than other.

operator fun dec(): Int

Decrements this value.

operator fun div(**other**: Byte): Int

operator fun div(**other**: Short): Int

operator fun div(**other**: Int): Int

operator fun div(**other**: Long): Long

Divides this value by the other value, truncating the result to an integer that is closer to zero.

fun equals(**other**: Int): Boolean

fun hashCode(): Int

Returns a hash code value for the object.

operator fun inc(): Int

Increments this value.

fun inv(): Int

Inverts the bits in this value.

operator fun minus(**other**: Byte): Int

operator fun minus(**other**: Short): Int

operator fun minus(**other**: Int): Int

operator fun minus(**other**: Long): Long

operator fun minus(**other**: Float): Float

operator fun minus(**other**: Double): Double

Subtracts the other value from this value.

infix fun or(**other**: Int): Int

Performs a bitwise OR operation between the two values.

operator fun plus(**other**: Byte): Int

operator fun plus(**other**: Short): Int

operator fun plus(**other**: Int): Int

operator fun plus(**other**: Long): Long

operator fun plus(**other**: Float): Float

operator fun plus(**other**: Double): Double

Adds the other value to this value.

operator fun rangeTo(**other**: Byte): [IntRange](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/-int-range/index.html)

operator fun rangeTo(**other**: Short): [IntRange](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/-int-range/index.html)

operator fun rangeTo(**other**: Int): [IntRange](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/-int-range/index.html)

operator fun rangeTo(**other**: Long): [LongRange](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/-long-range/index.html)

Creates a range from this value to the specified [other](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-int/range-to.html#kotlin.Int$rangeTo(kotlin.Byte)/other) value.

operator fun rem(**other**: Byte): Int

operator fun rem(**other**: Short): Int

operator fun rem(**other**: Int): Int

operator fun rem(**other**: Long): Long

operator fun rem(**other**: Float): Float

operator fun rem(**other**: Double): Double

Calculates the remainder of truncating division of this value by the other value.

The result is either zero or has the same sign as the *dividend* and has the absolute value less than the absolute value of the divisor.

infix fun shl(**bitCount**: Int): Int

Shifts this value left by the [bitCount](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-int/shl.html" \l "kotlin.Int$shl(kotlin.Int)/bitCount) number of bits.

infix fun shr(**bitCount**: Int): Int

Shifts this value right by the [bitCount](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-int/shr.html" \l "kotlin.Int$shr(kotlin.Int)/bitCount) number of bits, filling the leftmost bits with copies of the sign bit.

operator fun times(**other**: Byte): Int

operator fun times(**other**: Short): Int

operator fun times(**other**: Int): Int

operator fun times(**other**: Long): Long

operator fun times(**other**: Float): Float

operator fun times(**other**: Double): Double

Multiplies this value by the other value.

fun toByte(): Byte

Converts this [Int](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-int/index.html" \l "kotlin.Int) value to [Byte](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-byte/index.html#kotlin.Byte). **(+similar methods)**

operator fun unaryMinus(): Int

Returns the negative of this value.

operator fun unaryPlus(): Int

Returns this value.

infix fun ushr(**bitCount**: Int): Int

Shifts this value right by the [bitCount](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-int/ushr.html" \l "kotlin.Int$ushr(kotlin.Int)/bitCount) number of bits, filling the leftmost bits with zeros.

infix fun xor(**other**: Int): Int

Performs a bitwise XOR operation between the two values.

fun <T : [Comparable](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-comparable/index.html)<T>> T.coerceAtLeast(**minimumValue**: T): T

Ensures that this value is not less than the specified [minimumValue](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/coerce-at-least.html" \l "kotlin.ranges$coerceAtLeast(kotlin.ranges.coerceAtLeast.T,%20kotlin.ranges.coerceAtLeast.T)/minimumValue).

fun <T : [Comparable](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-comparable/index.html)<T>> T.coerceAtMost(**maximumValue**: T): T

Ensures that this value is not greater than the specified [maximumValue](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/coerce-at-most.html" \l "kotlin.ranges$coerceAtMost(kotlin.ranges.coerceAtMost.T,%20kotlin.ranges.coerceAtMost.T)/maximumValue).

fun <T : [Comparable](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-comparable/index.html)<T>> T.coerceIn(  
    **minimumValue**: T?,  
    **maximumValue**: T?  
): T

Ensures that this value lies in the specified range [minimumValue](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/coerce-in.html" \l "kotlin.ranges$coerceIn(kotlin.ranges.coerceIn.T,%20kotlin.ranges.coerceIn.T?,%20kotlin.ranges.coerceIn.T?)/minimumValue)..[maximumValue](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/coerce-in.html" \l "kotlin.ranges$coerceIn(kotlin.ranges.coerceIn.T,%20kotlin.ranges.coerceIn.T?,%20kotlin.ranges.coerceIn.T?)/maximumValue).

fun <reified R : Any> Byte.convert(): R

fun <reified R : Any> Short.convert(): R

fun <reified R : Any> Int.convert(): R

fun <reified R : Any> Long.convert(): R

fun <reified R : Any> [UByte](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-u-byte/index.html).convert(): R

fun <reified R : Any> [UShort](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-u-short/index.html).convert(): R

fun <reified R : Any> [UInt](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-u-int/index.html).convert(): R

fun <reified R : Any> [ULong](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-u-long/index.html).convert(): R

fun Byte.floorDiv(**other**: Byte): Int

fun Byte.floorDiv(**other**: Short): Int

fun Byte.floorDiv(**other**: Int): Int

fun Byte.floorDiv(**other**: Long): Long

Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.

fun Short.mod(**other**: Long): Long

fun Int.mod(**other**: Byte): Byte

fun Int.mod(**other**: Short): Short

fun Int.mod(**other**: Int): Int

Calculates the remainder of flooring division of this value by the other value.

infix fun Short.until(**to**: Byte): [IntRange](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/-int-range/index.html)

infix fun Char.until(**to**: Char): [CharRange](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/-char-range/index.html)

infix fun Int.until(**to**: Int): [IntRange](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/-int-range/index.html)

infix fun Long.until(**to**: Int): [LongRange](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/-long-range/index.html)

Returns a range from this value up to but excluding the specified [to](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/until.html" \l "kotlin.ranges$until(kotlin.Int,%20kotlin.Byte)/to) value.

If the [to](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.ranges/until.html" \l "kotlin.ranges$until(kotlin.Int,%20kotlin.Byte)/to) value is less than or equal to this value, then the returned range is empty.