

## رشته

کلاس رشته داخلی.

## شرح

است. از یونیکد پشتیبانی می کند و تمام GDScript این کلاس رشته داخلی (و کلاس مورد استفاده توسط ابزارهای لازم را برای مدیریت رشته ها فراهم می کند. رشته ها به صورت مرجع شمارش می شوند و از رویکرد کپی روی نوشتن استفاده می کنند، بنابراین انتقال آنها از نظر منابع ارزان است.

## آموزش ها

- [GDScript رشته های فرمت](#)

## مواد و روش ها

رشته	( رشته ) بوول از
رشته	( int from ) رشته
رشته	( رشته ) شناور از
رشته	( رشته ) بردار 2 از
رشته	( از Rect2 ) رشته
رشته	( رشته ) بردار 3 از
رشته	( از Transform2D ) رشته
رشته	( رشته ) صفحه از
رشته	( رشته ) کوات از
رشته	( از AABB ) رشته
رشته	( رشته ) پایه از
رشته	( رشته ) تبدیل از
رشته	( رشته ) رنگ از
رشته	( از NodePath ) رشته

رشته	( از RID ) رشته
رشته	( رشته ) فرهنگ لغت از
رشته	( رشته ) آرایه از
رشته	( از PoolByteArray ) رشته
رشته	( از PoolIntArray ) رشته
رشته	( از PoolRealArray ) رشته
رشته	( از PoolStringArray ) رشته
رشته	( از PoolVector2Array ) رشته
رشته	( از PoolVector3Array ) رشته
رشته	( از PoolColorArray ) رشته
بوول	( متن رشته ) starts_with
PoolStringArray	( ) بیگرام
رشته	c_escape ( )
رشته	c_unescape ( )
رشته	( ) بزرگ کردن
بین المللی	( رشته به ) casecmp_to
بین المللی	( رشته what، int from=0، int to=0 ) count
بین المللی	( رشته what، int from=0، int to=0 ) countn
رشته	( ) فرورفته
بوول	( ) خالی
بوول	( متن رشته ) ends_with
خالی	( int کاراکترهای ، int پاک کردن ) موقعیت
بین المللی	( رشته what، int from=0 ) پیدا کردن
بین المللی	( رشته چه ) find_last
بین المللی	( رشته what، int from=0 ) findn
رشته	( String placeholder ="{}" ، قالب ) مقادیر متغیر
رشته	get_base_dir ( )
رشته	get_basename ( )
رشته	get_extension ( )
رشته	get_file ( )

بین المللی	<code>()</code> هش
بین المللی	<code>hex_to_int ()</code>
رشته	<code>http_escape ()</code>
رشته	<code>http_unescape ()</code>
رشته	<code>humanize_size (اندازه داخلی)</code>
رشته	<code>()</code> تورفتگی (پیشوند رشته)
رشته	<code>int</code> درج (موقعیت)
بوول	<code>is_abs_path ()</code>
بوول	<code>is_rel_path ()</code>
بوول	<code>is_subsequence_of (متن رشته)</code>
بوول	<code>is_subsequence_ofi (متن رشته)</code>
بوول	<code>is_valid_filename ()</code>
بوول	<code>is_valid_float ()</code>
بوول	<code>is_valid_hex_number (bool with_prefix=false)</code>
بوول	<code>is_valid_html_color ()</code>
بوول	<code>is_valid_identifier ()</code>
بوول	<code>is_valid_integer ()</code>
بوول	<code>is_valid_ip_address ()</code>
رشته	<code>json_escape ()</code>
رشته	<code>()</code> سمت چپ (موقعیت درونی)
بین المللی	<code>()</code> طول
رشته	<code>lstrip (نویسه های رشته ای)</code>
بوول	<code>(String expr)</code> مطابقت
بوول	<code>matchn (string expr)</code>
PoolByteArray	<code>md5_buffer ()</code>
رشته	<code>md5_text ()</code>
بین المللی	<code>naturalnocasecmp_to (رشته به)</code>
بین المللی	<code>nocasecmp_to (رشته به)</code>
بین المللی	<code>ord_at (int at)</code>
رشته	<code>pad_decimals (int اعداد)</code>

رشته	pad_zeros ( اعداد بین‌المللی )
رشته	( ) درصد_رمزگشایی
رشته	( ) درصد_کد
رشته	plus_file ( فایل رشته ای )
رشته	( تکرار ) تعداد بین‌المللی
رشته	( جایگزین ) رشته چه، رشته برای چه
رشته	( String what, String forwhat ) جایگزین
بین‌المللی	rfind ( رشته what, int from=-1 )
بین‌المللی	rfindn ( رشته what, int from=-1 )
رشته	( سمت راست ) موقعیت درونی
PoolStringArray	rsplit ( جدیدکننده رشته , bool allow_empty=true, int maxsplit=0 )
رشته	rstrip ( نویسه های رشته ای )
PoolByteArray	sha1_buffer ( )
رشته	sha1_text ( )
PoolByteArray	sha256_buffer ( )
رشته	sha256_text ( )
شناور	( شباهت ) متن رشته
رشته	simplify_path ( )
PoolStringArray	split ( حداقل رشته , bool allow_empty=true, int maxsplit=0 )
PoolRealArray	split_floats ( جدیدکننده رشته , bool allow_empty=true )
رشته	strip_edges ( bool left=true, bool right=true )
رشته	strip_escapes ( )
رشته	substr ( int from, int len=-1 )
PoolByteArray	to_ascii ( )
شناور	to_float ( )
بین‌المللی	to_int ( )
رشته	to_lower ( )
رشته	to_upper ( )
PoolByteArray	to_utf8 ( )
PoolByteArray	to_wchar ( )

رشته	<code>trim_prefix</code> ( پیوند رشته )
رشته	<code>trim_suffix</code> ( پسوند رشته )
رشته	<code>validate_node_name</code> ( )
رشته	<code>xml_escape</code> ( )
رشته	<code>xml_unescape</code> ( )

## توضیحات روش

- ( رشته رشته ( بوول از

داده شده می سازد `bool` یک رشته جدید از

- ( `int` from ) رشته رشته

داده شده می سازد `int` یک رشته جدید از

- ( رشته رشته ( شناور از

داده شده می سازد `float` یک رشته جدید از

- ( رشته رشته ( بردار2 از

. داده شده می سازد `Vector2` یک رشته جدید از

- ( از `Rect2` ) رشته رشته

. داده شده می سازد `Rect2` یک رشته جدید از

- ( رشته رشته ( بردار 3 از

داده شده می سازد `Vector3` یک رشته جدید از

- ( از `Transform2D` ) رشته رشته

داده شده می سازد [Transform2D](#) یک رشته جدید از

---

- ( [رشته رشته](#) ) [صفحه](#) از

. یک رشته جدید از صفحه داده شده می سازد

---

- ( [رشته رشته](#) ) [کوات](#) از

داده شده می سازد [Quat](#) یک رشته جدید از

---

- ( از [AABB](#) ) [رشته رشته](#)

داده شده می سازد [AABB](#) یک رشته جدید از

---

- ( [رشته رشته](#) ) [پایه](#) از

یک رشته جدید از [پایه](#) داده شده می سازد

---

- ( [رشته رشته](#) ) [تبدیل](#) از

یک رشته جدید از [تبدیل](#) داده شده می سازد

---

- ( [رشته رشته](#) ) [رنگ](#) از

یک رشته جدید از [رنگ](#) داده شده می سازد

---

- ( از [NodePath](#) ) [رشته رشته](#)

. داده شده می سازد [NodePath](#) یک رشته جدید از

---

- ( از [RID](#) ) [رشته رشته](#)

داده شده می سازد [RID](#) یک رشته جدید از

- 
- ( رشته رشته ( فرهنگ لغت از

یک رشته جدید از دیکشنری داده شده می سازد

---

- ( رشته رشته ( آرایه از

یک رشته جدید از آرایه داده شده می سازد

---

- ( از PoolByteArray رشته رشته

. داده شده می سازد PoolByteArray یک رشته جدید از

---

- ( از PoolIntArray رشته رشته

. داده شده می سازد PoolIntArray یک رشته جدید از

---

- ( از PoolRealArray رشته رشته

. داده شده می سازد PoolRealArray یک رشته جدید از

---

- ( از PoolStringArray رشته رشته

. داده شده می سازد PoolStringArray یک رشته جدید از

---

- ( از PoolVector2Array رشته رشته

. داده شده می سازد PoolVector2Array یک رشته جدید از

---

- ( از PoolVector3Array رشته رشته

. داده شده می سازد PoolVector3Array یک رشته جدید از

---

- [رشته](#) ( از `PoolColorArray` )

. داده شده می سازد `PoolColorArray` یک رشته جدید از

- 
- [رشته](#) ( `bool starts_with` )

. با رشته داده شده شروع شود برمی گردد `true` اگر رشته

- 
- [بیگرم های](#) `PoolStringArray` ( )

. آرایه ای حاوی بیگرم (جفت حروف متوالی) این رشته را برمی گرداند

```
print("Bigrams".bigrams()) # Prints "[Bi, ig, gr, ra, am, ms]"
```

- 
- [رشته](#) `c_escape` ( )

. برمی گرداند C یک کپی از رشته را با نویسه های خاص با استفاده از استاندارد زبان

- 
- [رشته](#) `c_unescape` ( )

یک کپی از رشته را با نویسه های فراری که با معانی آنها جایگزین شده اند، برمی گرداند. دنباله های فرار `\v`، `\t`، `\r`، `\n`، `\f`، `\b`، `\a`، `\\`، `\?`، `\"`، `\'` پشتیبانی شده عبارتند از

. از دنباله فرار پشتیبانی نمی کند `\uXXXX` این روش GDScript، **توجه:** برخلاف تجزیه کننده

- 
- [حروف بزرگ](#) ( )

بزرگی برخی از حروف را تغییر می دهد. زیرخط را با فاصله جایگزین می کند، فاصله ها را قبل از حروف بزرگ در کلمه اضافه می کند، همه حروف را به حروف کوچک تبدیل می کند، سپس حرف اول و هر حرف بعد از یک `capitalize camelCase`. کاراکتر فاصله را بزرگ می کند. برای ، باز خواهد گشت

```
mixed_with_underscores Capitalize Camel Case Mixed With Underscores
```

- 
- [رشته به](#) ( `int casecmp_to` )



مقایسه حساس به حروف بزرگ را با رشته دیگری انجام می دهد. `1-` اگر کمتر از، اگر `1` بزرگتر از، یا `0` اگر مساوی باشد، برمی گرداند. "کمتر از" یا "بیشتر از" توسط `نقاط کد یونیکد` هر رشته تعیین می شود که تقریباً با ترتیب حروف الفبا مطابقت دارد.

رشته باشد یا `1-` اگر رشته «پایه» `to` رفتار با طول های رشته های مختلف: اگر `1` رشته «پایه» طولانی تر از رشته باشد، برمی گردد. به خاطر داشته باشید که این طول توسط تعداد کدهای یونیکد تعیین `to` کوتاه تر از می شود، نه کاراکترهای قابل مشاهده واقعی.

رشته خالی باشد یا `0` هر دو رشته `to` رفتار با رشته های خالی: اگر `1-` رشته «پایه» خالی باشد، `1` اگر خالی باشد، برمی گردد.

برای به دست آوردن نتیجه بولی از مقایسه رشته، به جای آن از `==` عملگر استفاده کنید. همچنین به `nocasecmp_to` مراجعه کنید .

- 
- `int` ( رشته what، `int` from=0، `int` to=0 ) تعداد

برابر با `0` `to` و `from` موقعیت ها را برمی گرداند. اگر `to` و `from` بین `what` تعداد تکرار رشته های فرعی برابر `0` باشد، از رشته فرعی باقی مانده استفاده می شود `to` باشد، کل رشته استفاده خواهد شد. اگر فقط

- 
- `int` countn ( رشته what، `int` from=0، `int` to=0 )

موقعیت ها را برمی `to` و `from` نادیده گرفتن حروف کوچک) بین `what` تعداد تکرار رشته های فرعی برابر `0` باشد، از رشته `to` برابر با `0` باشد، کل رشته استفاده خواهد شد. اگر فقط `to` و `from` گرداند. اگر فرعی باقی مانده استفاده می شود.

- 
- ( ) فرورفتگی رشته

یک کپی از رشته را با تورفتگی (برگه ها و فاصله های پیشرو) حذف می کند. همچنین به تورفتگی برای افزودن تورفتگی مراجعه کنید.

- 
- ( ) غول خالی

رشته برابر باشد برمی `0` گردد `true` اگر طول

- 
- `bool` ends\_with ( متن رشته )

به رشته داده شده ختم شود برمی گردد `true` اگر رشته

---

- `int` کاراکترهای ، `int` پاک کردن خالی ( موقعیت )

. `position` رشته را با شروع از `chars` کاراکترهای

---

- `int find` ( `what`، `int` from=0 رشته )

اولین رخداد یک زیررشته را پیدا می کند. موقعیت شروع رشته فرعی را برمی گرداند یا `-1` اگر پیدا نشد. به صورت اختیاری، فهرست جستجوی اولیه می تواند ارسال شود.

عملگر به `in` **توجه:** اگر فقط می خواهید بدانید که آیا یک رشته دارای یک رشته فرعی است یا خیر، از صورت زیر استفاده کنید:

```
# Will evaluate to `false`.
if "i" in "team":
    pass
```

- `int find_last` ( رشته چه )

آخرین رخداد یک زیررشته را پیدا می کند. موقعیت شروع رشته فرعی را برمی گرداند یا `-1` اگر پیدا نشد.

---

- `int findn` ( `what`، `int` from=0 رشته )

با نادیده گرفتن حروف، اولین رخداد یک زیررشته را پیدا می کند. موقعیت شروع رشته فرعی را برمی گرداند یا `-1` اگر پیدا نشد. به صورت اختیاری، فهرست جستجوی اولیه می تواند ارسال شود.

---

- `String placeholder = "{}"` ، قالب رشته ( مقادیر متغیر )

. `values` با `placeholder` رشته را با جایگزین کردن همه رخدادهای

---

- `get_base_dir` ( رشته )

اگر رشته یک مسیر فایل معتبر باشد، نام دایرکتوری پایه را برمی‌گرداند.

---

- **get\_basename ( )** رشته

اگر رشته یک مسیر فایل معتبر است، مسیر فایل کامل را بدون پسوند برمی‌گرداند.

---

- **get\_extension ( )** رشته

اگر رشته نام فایل یا مسیر معتبری باشد، پسوند را بدون کاراکتر نقطه اصلی ( ) برمی‌گرداند. اگر رشته `.` حاوی پسوند نباشد، به جای آن یک رشته خالی برمی‌گرداند.

```
print("/path/to/file.txt".get_extension()) # "txt"
print("file.txt".get_extension()) # "txt"
print("file.sample.txt".get_extension()) # "txt"
print(".txt".get_extension()) # "txt"
print("file.txt.".get_extension()) # "" (empty string)
print("file.txt..".get_extension()) # "" (empty string)
print("txt".get_extension()) # "" (empty string)
print("".get_extension()) # "" (empty string)
```

- **get\_file ( )** رشته

اگر رشته یک مسیر فایل معتبر است، نام فایل را برمی‌گرداند.

---

- **int ( )** هش

مقدار هش 32 بیتی را که نشان دهنده محتوای رشته است برمی‌گرداند.

با محتوای برابر همیشه مقادیر هش یکسان تولید می‌کند. با این حال، برعکس آن درست `String` **توجه:** نیست. برگرداندن مقادیر هش یکسان به معنای برابری رشته‌ها نیست، زیرا رشته‌های مختلف به دلیل برخورد هش می‌توانند مقادیر هش یکسانی داشته باشند.

---

- **int hex\_to\_int ( )**

یک رشته حاوی یک عدد هگزادسیمال را به یک عدد صحیح تبدیل می کند. انتظار می رود رشته های `0x` برگردانده می شود "0" هگزادسیمال با "0" پیشوند شوند در غیر این صورت `0`.

```
print("0xff".hex_to_int()) # Print "255"
```

- **رشته http\_escape ( )**

نامیده می "URL دوستانه فرار می کند (کد می کند). همچنین به عنوان "کد URL از یک رشته به فرمت شود.

```
print("https://example.org/?escaped=" + "Godot Engine:'docs'".http_escape())
```

- **رشته http\_unescape ( )**

کدگذاری شده حذف می کند (رمزگشایی می کند). همچنین به عنوان "رمزگشایی URL یک رشته را در قالب URL نیز نامیده می شود.

```
print("https://example.org/?escaped=" +  
"Godot%20Engine%3A%27docs%27".http_unescape())
```

- **رشته humanize\_size ( int اندازه )**

، `size` B، KiB، MiB، GiB، TiB، PiB، با استفاده از مجموعه ای بین المللی از واحدهای اندازه داده، یعنی به صورت تعداد بایت به قالب قابل خواندن برای انسان تبدیل می شود. توجه داشته باشید که Eib، کوچکترین واحد بعدی به طور خودکار برای نگهداری حداکثر 1024 واحد انتخاب می شود.

```
var bytes = 133790307  
var size = String.humanize_size(bytes)  
print(size) # prints "127.5 MiB"
```

- (تورفتگی رشته) پیشوند رشته

. `prefix` یک کپی از رشته را با خطوط فرورفته با

یا چهار فاصله با استفاده از تورفتگی کرد `"\t\t"` به عنوان مثال، رشته را می توان با دو زبانه با استفاده از . پیشوند می تواند هر رشته ای باشد، بنابراین می توان از آن برای نظر دادن رشته ها با استفاده از مثال `"# "` برای حذف تورفتگی مراجعه کنید `dedent` استفاده کرد. همچنین به

.توجه: خطوط خالی خالی نگه داشته می شوند

---

- ( what String ، int درج رشته ) موقعیت

.درج شده در موقعیت داده شده برمی گرداند `what` یک کپی از رشته را با رشته فرعی

---

- `bool is_abs_path ( )`

.اگر مسیر مطلق باشد، برمی گردد `true`، اگر رشته مسیری به یک فایل یا دایرکتوری باشد

---

- `bool is_rel_path ( )`

.اگر مسیر نسبی باشد، برمی گردد `true`، اگر رشته مسیری به یک فایل یا دایرکتوری باشد

---

- `bool is_subsequence_of ( متن رشته )`

.ای از رشته داده شده باشد برمی گردد `true` اگر این رشته دنباله

---

- `bool is_subsequence_ofi ( متن رشته )`

.ای از رشته داده شده باشد، بدون در نظر گرفتن حروف کوچک، برمی گردد `true` اگر این رشته دنباله

---

- `bool is_valid_filename ( )`

:باشد که در نام فایل ها مجاز نیستند، برمی گردد `true` اگر این رشته فاقد کاراکترهایی

: / \ ? \* " | % < >

- **bool is\_valid\_float ( )**

این رشته حاوی یک شناور معتبر باشد، برمی گردد. این شامل اعداد صحیح است و از نماها نیز `true` اگر پشتیبانی می کند:

```
print("1.7".is_valid_float()) # Prints "True"
print("24".is_valid_float()) # Prints "True"
print("7e3".is_valid_float()) # Prints "True"
print("Hello".is_valid_float()) # Prints "False"
```

- **bool is\_valid\_hex\_number ( bool with\_prefix=false )**

، `true` باشد `with_prefix` معتبر باشد، برمی گردد. اگر `true` اگر این رشته دارای یک عدد هگزادسیمال `xDEADC0DE` پیشوند تعیین می شود، به عنوان مثال: `0x` اعتبار عدد هگزادسیمال با `0`

- **bool is\_valid\_html\_color ( )**

HTML هگزادسیمال باشد، برمی گردد. سایر نمادهای HTML این رشته دارای رنگ معتبر در نماد `true` اگر شده توسط این روش معتبر تلقی نمی شوند و برمی گردند `hsl()` مانند رنگ ها یا رنگ های نام گذاری `false`.

- **bool is\_valid\_identifier ( )**

این رشته یک شناسه معتبر باشد برمی گردد. یک شناسه معتبر ممکن است فقط شامل حروف، `true` اگر اعداد و زیرخط ( `_` ) باشد و اولین کاراکتر ممکن است یک رقم نباشد.

```
print("good_ident_1".is_valid_identifier()) # Prints "True"
print("1st_bad_ident".is_valid_identifier()) # Prints "False"
print("bad_ident_#2".is_valid_identifier()) # Prints "False"
```

- `bool is_valid_integer ( )`

این رشته دارای یک عدد صحیح معتبر باشد، برمی گردد `true` اگر

```
print("7".is_valid_int()) # Prints "True"
print("14.6".is_valid_int()) # Prints "False"
print("L".is_valid_int()) # Prints "False"
print("+3".is_valid_int()) # Prints "True"
print("-12".is_valid_int()) # Prints "True"
```

---

- `bool is_valid_ip_address ( )`

با قالب بندی مناسب باشد، برمی گردد. این روش IPv4 یا IPv6 این رشته فقط دارای یک آدرس `true` اگر  
رزرو شده مانند `0.0.0.0` معتبر را در نظر می گیرد IP آدرس های

---

- `json_escape ( )` رشته

فرار کرده اند، برمی گرداند JSON یک کپی از رشته را با کاراکترهای خاص که با استفاده از استاندارد

---

- `سمت چپ ( موقعیت درونی )` رشته

تعدادی کاراکتر را از سمت چپ رشته برمی گرداند.

---

- `int ( )` طول

تعداد کاراکترهای رشته را برمی گرداند.

---

- `lstrip ( )` رشته ای ( نویسه های رشته ای )

ای است که `chars` یک کپی از رشته را با کاراکترهای حذف شده از سمت چپ برمی گرداند. آرگومان رشته  
مجموعه کاراکترهایی را که باید حذف شوند را مشخص می کند.

را ببینید که یک رشته پیشوند را به جای مجموعه ای `trim_prefix` یک پیشوند نیست. `chars` `روش` The `توجه`  
از کاراکترها حذف می کند.

- **Bool match** ( *String* expr )

آیا یک عبارت ساده حساس به حروف بزرگ مطابقت دارد، جایی که `"**"` با صفر یا چند کاراکتر دلخواه مطابقت دارد و `"?"` با هر کاراکتری به جز نقطه ( `"."` ) مطابقت دارد. یک رشته یا عبارت خالی همیشه به `false`.

- **bool matchn** ( *string* expr )

آیا یک عبارت ساده غیر حساس به حروف بزرگ مطابقت دارد، جایی که `"**"` با صفر یا چند کاراکتر دلخواه مطابقت دارد و `"?"` با هر کاراکتری به جز نقطه ( `"."` ) مطابقت دارد. یک رشته یا عبارت خالی همیشه به `false`.

- **PoolByteArray md5\_buffer** ( )

رشته را به صورت آرایه ای از بایت برمی گرداند MD5 هش

- **md5\_text** ( ) *رشته*

رشته را به صورت رشته برمی گرداند MD5 هش

- **int naturalnocasecmp\_to** ( *رشته* به )

*مقایسه ترتیب طبیعی* غیر حساس به حروف بزرگ را با رشته دیگری انجام می دهد. `-1` اگر کمتر از، اگر `1` بزرگتر از، یا `0` اگر مساوی باشد، برمی گرداند. "کمتر از" یا "بیشتر از" توسط *نقاط کد یونیکد* هر رشته تعیین می شود که تقریباً با ترتیب حروف الفبا مطابقت دارد. در داخل، کاراکترهای کوچک در طول مقایسه به حروف بزرگ تبدیل می شوند.

هنگامی که برای مرتب سازی استفاده می شود، مقایسه ترتیب طبیعی مجموعه هایی از اعداد را همانطور که اکثر مردم انتظار دارند ترتیب می دهد. اگر اعداد را از 1 تا 10 با ترتیب طبیعی مرتب کنید، به جای `[1, 2, 3, ...]` `[1, 10, 2, 3, ...]`

رشته باشد یا `-1` اگر رشته «پایه» `to` رفتار با طول های رشته های مختلف: اگر `1` رشته «پایه» طولانی تر از رشته باشد، برمی گردد. به خاطر داشته باشید که این طول توسط تعداد کدهای یونیکد تعیین `to` کوتاه تر از می شود، نه کاراکترهای قابل مشاهده واقعی.



رشته خالی باشد یا `0` هر دو رشته `to` رفتار با رشته های خالی: اگر `1-` رشته «پایه» خالی باشد، `1` اگر خالی باشد، برمی گردد.

برای به دست آوردن نتیجه بولی از مقایسه رشته، به جای آن از `==` عملگر استفاده کنید. همچنین به [مراجعه](#) کنید `casecmp_to` و `nocasecmp_to`.

---

- `int nocasecmp_to` ( رشته به )

مقایسه‌ای را با رشته‌ای دیگر انجام می‌دهد. `1-` اگر کمتر از، اگر `1` بزرگتر از، یا `0` اگر مساوی باشد، برمی‌گرداند. "کمتر از" یا "بیشتر از" توسط [نقاط کد یونیکد](#) هر رشته تعیین می‌شود که تقریباً با ترتیب حروف الفبا مطابقت دارد. در داخل، کاراکترهای کوچک در طول مقایسه به حروف بزرگ تبدیل می‌شوند.

رشته باشد یا `1-` اگر رشته «پایه» `to` رفتار با طول‌های رشته‌های مختلف: اگر `1` رشته «پایه» طولانی‌تر از رشته باشد، برمی‌گردد. به خاطر داشته باشید که این طول توسط تعداد کدهای یونیکد تعیین `to` کوتاه‌تر از می‌شود، نه کاراکترهای قابل مشاهده واقعی.

رشته خالی باشد یا `0` هر دو رشته `to` رفتار با رشته های خالی: اگر `1-` رشته «پایه» خالی باشد، `1` اگر خالی باشد، برمی گردد.

برای به دست آوردن نتیجه بولی از مقایسه رشته، به جای آن از `==` عملگر استفاده کنید. همچنین به [مراجعه](#) کنید `casecmp_to` را ببینید.

---

- `int ord_at ( int at )`

`at` کد کاراکتر را در موقعیت برمی‌گرداند.

---

- `int` رشته ای ( اعداد `pad_decimals` )

بعد از نقطه اعشار داشته باشد `digits` یک عدد را طوری قالب بندی می‌کند که یک عدد دقیق

---

- `int` ارقام ( `pad_zeros` رشته )

قبل از نقطه اعشار باشد `digits` یک عدد را طوری قالب بندی می‌کند که عدد دقیق آن

---

- ( ) رشته درصد\_رمزگشایی

. را ببینید `encode` یک رشته رمزگذاری شده درصد را رمزگشایی کنید. درصد

---

- `encode` رشته درصد ( )

با فرم POST و بدنه درخواست‌های HTTP GET درصد یک رشته را رمز می‌کند. هنگام ارسال درخواست رمزگذاری می‌کند URL پارامترها را در (urlencoded).

---

- `plus_file` رشته ( فایل رشته )

در انتهای رشته به عنوان یک مسیر فرعی به هم متصل می‌شود. به عنوان `file`، اگر رشته یک مسیر باشد مثال `"this/is".plus_file("path") == "this/is/path"`

---

- `int` تکرار رشته ( تعداد )

رشته اصلی چند بار تکرار شده را برمی‌گرداند. تعداد تکرارها با استدلال داده می‌شود.

---

- `String` what, `String` forwhat ( جایگزینی رشته )

وقوع یک زیررشته حساس به حروف بزرگ را با رشته داده شده در داخل رشته جایگزین می‌کند.

---

- `String` جایگزین ( رشته چه، رشته برای چه )

وقوع یک زیررشته حساس به حروف بزرگ را با رشته داده شده در داخل رشته جایگزین می‌کند.

---

- `int` rfind ( رشته what، `int` from=-1 )

یک جستجوی حساس به حروف بزرگ برای یک زیررشته انجام می‌دهد، اما از انتهای رشته به جای ابتدا شروع می‌شود.

---

- `int` rfindn ( رشته what، `int` from=-1 )

یک جستجوی بدون حروف بزرگ برای یک رشته فرعی انجام می‌دهد، اما از انتهای رشته به جای ابتدا شروع می‌شود.

---

- ( رشته سمت راست ) موقعیت درونی

سمت راست رشته را از یک موقعیت مشخص برمی گرداند.

---

- PoolStringArray rsplit ( جدیدکننده رشته ، bool allow\_empty=true, int maxsplit=0 )

رشته تقسیم می کند و آرایه ای از رشته های فرعی را که از سمت راست شروع `delimiter` رشته را با یک می شود، برمی گرداند.

تقسیمات در آرایه برگشتی به همان ترتیب رشته اصلی، از چپ به راست مرتب می شوند.

مشخص شده باشد، تعداد تقسیم‌هایی را که باید از راست به بالا انجام شود را `maxsplit` اگر مشخص می‌کند. مقدار پیش‌فرض 0 به این معنی است که همه موارد تقسیم می‌شوند، بنابراین `maxsplit` . نتیجه مشابه تقسیم می‌شود

مثال:

```
var some_string = "One,Two,Three,Four"
var some_array = some_string.rsplit(",", true, 1)
print(some_array.size()) # Prints 2
print(some_array[0]) # Prints "One,Two,Three"
print(some_array[1]) # Prints "Four"
```

- ( کاراکترهای رشته ) `rstrip` رشته

ای است که `chars` یک کپی از رشته را با کاراکترهای حذف شده از سمت راست برمی گرداند. آرگومان رشته مجموعه کاراکترهایی را که باید حذف شوند را مشخص می کند.

را ببینید که یک رشته پسوند منفرد را به جای مجموعه `trim_suffix` پسوند نیست. روش `chars` The `chars`:توجه ای از کاراکترها حذف می کند.

---

- PoolByteArray sha1\_buffer ( )

رشته را به صورت آرایه ای از بایت برمی گرداند SHA-1 هش.

---

- [رشته sha1\\_text \( \)](#)

رشته را به صورت رشته برمی گرداند SHA-1 هش

---

- [PoolByteArray sha256\\_buffer \( \)](#)

رشته را به صورت آرایه ای از بایت برمی گرداند SHA-256 هش

---

- [رشته sha256\\_text \( \)](#)

رشته را به صورت رشته برمی گرداند SHA-256 هش

---

- [شبهات شناور \( متن رشته \)](#)

این رشته را در مقایسه با رشته دیگر برمی گرداند. نتیجه 1.0 به ( [Sorensen-Dice](#) شاخص شبهات ( [ضریب](#) معنای کاملاً مشابه است، در حالی که 0.0 به معنای کاملاً متفاوت است.

```
print("ABC123".similarity("ABC123")) # Prints "1"
print("ABC123".similarity("XYZ456")) # Prints "0"
print("ABC123".similarity("123ABC")) # Prints "0.8"
print("ABC123".similarity("abc123")) # Prints "0.4"
```

- [رشته simplify\\_path \( \)](#)

یک مسیر متعارف ساده شده را برمی گرداند

---

- [PoolStringArray \(تخمین کننده رشته ، bool allow\\_empty=true, int maxsplit=0 تقسیم](#)

رشته تقسیم می کند و آرایه ای از رشته های فرعی را برمی گرداند. می `delimiter` رشته را توسط یک `delimiter` تواند از هر طولی باشد.

مشخص شده باشد، تعداد تقسیم‌هایی را که باید از چپ به بالا انجام شود را `maxsplit` اگر مشخص می‌کند. مقدار پیش فرض `0` به این معنی است که همه موارد تقسیم می شوند `maxsplit`

مثال:

```
var some_string = "One,Two,Three,Four"
var some_array = some_string.split(",", true, 1)
print(some_array.size()) # Prints 2
print(some_array[0]) # Prints "One"
print(some_array[1]) # Prints "Two,Three,Four"
```

استفاده کنید RegEx اگر نیاز به تقسیم رشته ها با قوانین پیچیده تر دارید، به جای آن از کلاس

- `PoolRealArray split_floats` (جدیدکننده رشته ، `bool allow_empty=true`)

رشته را با استفاده از یک رشته جداکننده به شناور تقسیم می کند و آرایه ای از رشته های فرعی را برمی گرداند.

، "،" برای مثال، "1,2.5,3" در [1,2.5,3] صورت تقسیم بر

- `String strip_escapes` ( `bool left=true`, `bool right=true` ) نوار لبه های رشته

یک کپی از رشته را که فاقد هر گونه کاراکتر غیرقابل چاپ (شامل جدول ها، فاصله ها و شکسته های خط) در در لبه های چپ stripping ابتدا و انتهای آن است، برمی گرداند. آرگومان های اختیاری به ترتیب برای جابجایی و راست استفاده می شود.

- `String strip_escapes` ( )

یک کپی از رشته بدون هر گونه کاراکتر فرار را برمی گرداند. اینها شامل تمام نویسه های کنترلی غیرقابل ( `\r` و `\n` ) و نویسه های خط جدید (C در `\t` ) مانند جدول، (< 32) ASCII چاپ صفحه اول جدول می شود، اما فاصله ها نیست.

- `String strip_escapes` ( `int from`, `int len=-1` ) رشته فرعی

اختیاری است و استفاده از `len` آرگومان. `len` با طول برمی گرداند `from` بخشی از رشته را از موقعیت `1-` آن کاراکترهای باقی مانده را از موقعیت داده شده برمی گرداند.

- **PoolByteArray to\_ascii ( )**

که آرایه ای از بایت است) تبدیل می کند. این تبدیل **PoolByteArray** رشته (که یک آرایه کاراکتری است) را به سریعتر است ، زیرا در این روش فرض می شود که تمام کاراکترهای رشته، کاراکترهای **to\_utf8** در مقایسه با ASCII هستند.

---

- **float to\_float ( )**

این روش روی اولین کاراکتر غیر عددی به **float** یک رشته حاوی یک عدد اعشاری را به یک تبدیل می کند که برای نمایی استفاده می شود، متوقف می شود **e**، (جز اولین **.** (نقطه اعشاری

```
print("12.3".to_float()) # 12.3
print("1.2.3".to_float()) # 1.2
print("12ab3".to_float()) # 12
print("1e3".to_float()) # 1000
```

- **int to\_int ( )**

این روش هر کاراکتر غیر عددی را حذف **int** یک رشته حاوی یک عدد صحیح را به یک تبدیل می کند **.** می کند و در صورت مواجهه با یک کاراکتر متوقف می شود **.**

```
print("123".to_int()) # 123
print("a1b2c3".to_int()) # 123
print("1.2.3".to_int()) # 1
```

- **رشته to\_lower ( )**

رشته تبدیل شده به حروف کوچک را برمی گرداند.

---

- **رشته به بالا ( )**

رشته تبدیل شده به حروف بزرگ را برمی گرداند.

---

- **PoolByteArray to\_utf8 ( )**

که آرایه ای از بایت است) تبدیل می کند. تبدیل PoolByteArray رشته (که آرایه ای از کاراکترها است) را به پشتیبانی می کند. بنابراین، شما باید این تابع را UTF-8 است، اما از همه کاراکترهای to\_ascii کمی کندتر از . ترجیح دهید to\_ascii به

---

- **PoolByteArray to\_wchar ( )**

که آرایه ای از بایت است) تبدیل می کند) PoolByteArray رشته (که آرایه ای از کاراکترها است) را به

---

- **trim\_prefix ( رشته )** (پیشوند رشته)

اگر رشته ای با آن شروع شود یا رشته را بدون تغییر بگذارد، از ابتدا حذف می کند.

---

- **trim\_sffix ( رشته )** (پسوند رشته)

اگر رشته ای به آن ختم شود یا رشته را بدون تغییر بگذارد، از انتهای آن حذف می کند.

---

- **validate\_node\_name ( رشته )**

هر کاراکتر را از رشته که در نام گره ها ممنوع است ( " / @ : . ) حذف می کند

---

- **xml\_escape ( رشته )**

فرار کرده اند، برمی گرداند XML یک کپی از رشته را با کاراکترهای خاص که با استفاده از استاندارد

---

- **xml\_unescape ( رشته )**

جایگزین شده اند، برمی XML یک کپی از رشته را با کاراکترهای فراری که با معانی آنها مطابق استاندارد گرداند.