No.	SubSig	APIPair	-	RegexWrapper	-	-	-	
		API Invocation	CondExpr	R	L	S	Lmin	Relation
1	boolean isEmpty()	caller: str array r=str.isEmpty()	r==T	[\s\S]{0}	-1	NULL	0	=
-	boolean equals(Object)	r=str.equals(v)	r==T	V	-1	NULL	Len(v)	=
	int compareTo(String)	r=str.compareTo(v)	r==0	V	-1	NULL	Len(v)	=
	int compareTo(Object) boolean contentEquals(CharSequence)	r=str.compareTo(v) r=str.contentEquals(v)	r==0 r==T	v v	-1 -1	NULL NULL	Len(v) Len(v)	≡
	boolean contentEquals(StringBuffer)	r=str.contentEquals(v)	r==0	V	-1	NULL	Len(v)	⊈
	int compareToIgnoreCase(String)	r=str.compareToIgnoreCase(v)	r==0	V	-1	NULL	Len(v)	⊈
	int boolean equalsIgnoreCase(String) boolean contains(CharSequence)	r=str.equalsIgnoreCase(v) r=str.contains(v)	r==0 r==T	v v[\s\S]*	-1 1	NULL NULL	Len(v) Len(v)	⊈
	boolean startsWith(String)	r=str.startsWith(v)	r==T	v[\s\S]*	-1	NULL	Len(v)	≡
	boolean startsWith(String,int)	r=str.startsWith(v,i ₁)	r==T	[\s\S]{i ₁ }v[\s\S]*	-1	NULL	i_1 +Len(v)	≡
12	boolean endsWith(String)	r=str.endsWith(v)	r==T	[\s\S]*v	-1	NULL	Len(v)	=
12	int in land (Stain a)	r=str.indexOf(v)	r==i ₁	$[^{v}]\{i_{1}\}v[^{s}]^{*}$	-1	NULL	i_1 +Len(v)	
13	int indexOf(String)	r=str.indexOf(v) r=str.indexOf(v)	r>=i ₁	$[^{v}]\{i_{1},\}v[^{s}]^{*}$ $[^{v}]\{0,;\}v[^{s}]^{*}$	-1 1	NULL NULL	i ₁ +Len(v) Len(v)	⊈
		r=str.indexOf(v)	$r \le i_1$ $r = i_1$		-1 -1	NULL	i_1 +Len(v)	_
1.4		r=str.indexOf(v)	$r>=i_1$	$[v]\{i_1, v[s] $	-1	NULL	i_1 +Len(v)	_
14		r=str.indexOf(v)	$r \le i_1$	$[^{v}]\{0,i_{1}\}v[^{s}]^{*}$	-1	NULL	Len(v)	=
				(convert v to string first)				
1.5	int indexOf(String,int)	r=str.indexOf(v , i_1)	$r==i_2$	$[\s\S]\{i_1\}[\v]\{i_2-i_1\}\v[\s\S]^*$	-1	NULL	i_2 +Len(v)	1
15		r=str.indexOf(v,i ₁)	r>=i ₂	$[\s\S]\{i_1\}[\v]\{i_2-i_1,\}\v[\s\S]^*$	-l	NULL	i_2 +Len(v)	1
		r=str.indexOf(v,i ₁) r=str.indexOf(v,i ₁)	$r \le i_2$ $r = i_2$	$[\s\S]\{i_1\}[\v]\{0,i_2-i_1\}v[\s\S]^*$ $[\s\S]\{i_1\}[\v]\{i_2-i_1\}v[\s\S]^*$	-1 _1	NULL NULL	i_1 +Len(v) i_2 +Len(v)	+
	int indexOf(int,int)	$r=\text{str.indexOf}(v,i_1)$	$r > = i_2$		-1	NULL	i_2 +Len(v)	
16		$r=str.indexOf(v,i_1)$	$r <= i_2$	$[\S \S] \{i_1\} [\S \S] \{0, i_2 - i_1\} v [\S \S] *$	-1	NULL	i_1 +Len(v)	
		(· · · · ·	_	(convert v to string first)				
1.5	int lastIndexOf(String)	r=str.lastIndexOf(v)	r==i ₁	$[\s\s]\{i_1\}v[\v]^*$	-1	NULL	i_1 +Len(v)	
17		r=str.lastIndexOf(v)	r>=i ₁	[\s\S]{i ₁ ,}v[^v]*	-1	NULL	i_1 +Len(v)	
		r=str.lastIndexOf(v) r=str.lastIndexOf(v)	$r \le i_1$ $r = i_1$	$ [\s\s] \{0,i_1\} v [\v]^* $ $ [\s\s] \{i, \v\s]^* $	-1 -1	NULL NULL	$\frac{\text{Len(v)}}{i_1 + \text{Len(v)}}$	⊈
10	int lastIndexOf(int)	r=str.lastIndexOf(v) r=str.lastIndexOf(v)	$r==i_1$ $r>=i_1$		-1 -1	NULL	i_1 +Len(v) i_1 +Len(v)	
18		r=str.lastIndexOf(v)	$r <= i_1$	$[\s\s]\{0,i_1\}v[\v]^*$	-1	NULL	Len(v)	=
		()	-	(convert v to string first)			\ /	
	int lastIndexOf(String,int)	r=str.lastIndexOf(v,i ₁)	$r==i_2$	$[\s\s]\{i_2\}v[\v]\{i_1+1-i_2-Len(v)\}[\s\s]^*$	-1	NULL	i ₁ +1	⊈
19		r=str.lastIndexOf(v, i_1)	r>=i ₂	$[\s\] \{i_2\} \ v[\v] \{i_1+1-i_2-Len(v)\} [\s\]^*$	-1	NULL	i ₁ +1	⊈
		r=str.lastIndexOf(v,i ₁) r=str.lastIndexOf(v,i ₁)	$r \le i_2$ $r = i_2$		-1 _1	NULL NULL	i_1+1 i_1+1	⊈
20	int lastIndexOf(int,int)	r-str.lastIndexOf(v,i ₁) r=str.lastIndexOf(v,i ₁)	$r = i_2$ $r > = i_2$	$[\s\s]\{i_2\}\v[\v]\{i_1+1-i_2-Len(v)\}[\s\s]^*$	-1 -1	NULL	i_1+1	= ⊈
	int tastificaci (int,int)	$r=str.lastIndexOf(v,i_1)$	$r < i_2$	$[\s\s]\{i_2\}\v[\v]\{i_1+1-i_2-Len(v)\}[\s\s]^*$	-1	NULL	i_1+1	⊈
21	boolean matches(String)	r=str.matches(v)	r==T	V	-1	NULL	Len(v)	=
22	boolean regionMatches	r=str.regionMatches(i_1, v_1, i_2, i_3)	r==T	$[\s\S]\{i_1\}subStr[\s\S]*$	-1	NULL	i_1+i_3	=
	(int,String,int,int)	1 surregioniviationes(1], v 1,12,13)	1 1	$subStr=v_1.substring(i_2,i_2+i_3);$	-1	NOLL	11 + 13	
73	boolean regionMatches	r=str.regionMatches(b ₁ ,i ₁ ,v ₁ ,i ₂ ,i ₃)	r==T	$[\s\S]\{i_1\}$ subStr $[\s\S]$ *	-1	NULL	i_1+i_3	⊈
	(boolean,int,String,int,int)			$subStr=v_1.substring(i_2,i_2+i_3);$				
24	char charAt(int)	r=str.charAt(i ₁)	r==i ₁	$[\s\S]\{i_1\}v[\s\S]^*$	-1	NULL	i_1 +Len(v)	
24		r=str.charAt(i ₁)	r>=i ₁	[\s\S]{i ₁ ,}v[\s\S]*	- l 1	NULL NULL	i_1 +Len(v)	
	int codePointAt(int)	r=str.charAt(i_1) r=str.codePointAt(i_1)	$r \le i_1$ $r = i_1$		-1 -1	NULL	$\frac{\text{Len(v)}}{i_1 + \text{Len(v)}}$	≢
25		r=str.codePointAt(i ₁)	$r>=i_1$	[\s\S]{i ₁ ,}v[\s\S]*	-1	NULL	i_1 +Len(v)	
		r=str.codePointAt(i ₁)	$r <= i_1$	[\s\S]{0,i ₁ }v[\s\S]*	-1	NULL	Len(v)	⊈
		r=str.codePointCount()	$r==i_1$	$[\s\]\{i_1\}$	-1	NULL	i_1	≡
26	int codePointCount(int,int)	r=str.codePointCount()	r>=i ₁	$[\s\S]\{i_1,\}$	-1	NULL	i_1	⊈
		r=str.codePointCount()	r<=i1	$[\s]\{0,i_1\}$	-1	NULL	0	⊈
	int codePointBefore(int)	r=str.codePointBefore(i ₁) r=str.codePointBefore(i ₁)	$r==i_2$	$[\s\s]\{i_1-1\}i_2[\s\s]^*$	-1 1	NULL NULL	1 ₁	≡
27		r=str.codePointBefore(i ₁) r=str.codePointBefore(i ₁)	$r>=i_2$ $r<=i_2$	$[\s\S]\{i_1-1\}i_2[\s\S]^*$ $[\s\S]\{i_1-1\}i_2[\s\S]^*$	-1 -1	NULL	1 ₁	⊈
			1 \ 12	(convert i_2 -1, i_2 +1, i_2 to string first)	1	IVOLL	1	<i>=</i>
20	·		$r==i_1$		1	NILIT		<i>d</i>
28	int offsetByCodePoints(int,int)	r=str.offsetByCodePoints(i ₁ ,i ₂)	(>=,<=,>,<)	$[\slash S]\{i_2\}$	-1	NULL	12	⊈
	int length()	r=str.length()	$r==i_1$	$[\s\S]\{i_1\}$	-1	NULL	i_1	≡
29		r=str.length()	r>=i ₁	[\s\S]{i ₁ ,}	-1	NULL	i ₁	=
		r=str.length() r=str.toCharArray()	r<=i ₁		-1 -1	NULL NULL	0 i+Len(v)	=
20	char[] toCharArray()	r=str.toCharArray() r=str.toCharArray()	r[i]==v r[i]>v	[\s\S]{i}{V\1][\s\S]*	-1 -1	NULL	i+Len(v)	
30		r=str.toCharArray()	r[i] <v< td=""><td>[\s\S]{i}[v-1][\ s\S]*</td><td>-1</td><td>NULL</td><td>i+Len(v)</td><td>⊈</td></v<>	[\s\S]{i}[v-1][\ s\S]*	-1	NULL	i+Len(v)	⊈
		an akan at Daya ()		(convert v+1,v-1 to string first)	1	ATT TF T	*: T	
		r=str.getBytes(v ₁)		[\s\S]{i}v[\s\S]* [\s\S]{i}(v+1)[\s\S]*	-1 -1	NULL NULL	()	≡
31	byte[] getBytes(String)	r=str.getBytes(v ₁) r=str.getBytes(v ₁)		[\s\S]{i}[v+1][\s\S]* [\s\S]{i}[v-1][\s\S]*	-1 -1	NULL	` ′	⊈
		o	- [-] ''	(convert v+1,v-1 to string first)			- ===(<i>\(\)</i>	-
		r=str.getBytes(v ₁)	r[i]==v	[\s\S]{i}v[\s\S]*	-1	NULL	i+Len(v)	≡
32	byte[] getBytes(Charset)	r=str.getBytes(v ₁)	r[i]>v	$[\s\s] \{i\} [v+1] [\s\s]^*$	-1	NULL	` ′	
		r =str.getBytes(v_1)	r[i] <v< td=""><td>[\s\S]{i}[v-1][\s\S]*</td><td>-1</td><td>NULL</td><td>i+Len(v)</td><td>⊈</td></v<>	[\s\S]{i}[v-1][\s\S]*	-1	NULL	i+Len(v)	⊈
		r=str.getBytes()	r[i]==v	(convert v+1,v-1 to string first) $[\s\]\{i\}v[\s\S]^*$	-1	NULL	i+Len(v)	=
33	hvte[] getRvtes()	r=str.getBytes()	r[i]>v	[\s\S]{i}[v+1][\s\S]*	-1	NULL	i+Len(v)	⊈
JJ	byte[] getBytes()	r=str.getBytes()	r[i] <v< td=""><td>[\s\S]{i}[v-1][\s\S]*</td><td>-1</td><td>NULL</td><td>` ′</td><td></td></v<>	[\s\S]{i}[v-1][\s\S]*	-1	NULL	` ′	
2.4	Chairman 1 at 1 C A		NIT IT T	(convert v+1,v-1 to string first)	1	ATT TF T		
	String substring(int) String substring(int int)	r=str.substring(i ₁)	NULL NULL	[\s\S]{i ₁ } [\s\S]{i ₁ }	-1 i ₂ -i ₁	NULL NULL	1 ₁	≡
	String substring(int,int) CharSequence subSequence(int,int)	r=str.substring(i ₁ ,i ₂) r=str.subSequence(i 1,i 2)	NULL		i_2 - i_1	NULL	11 11	⊈
	String[] split(String)	r=str.split(v)		[v]{i}	<u>-</u> 2 1	[^v]	i	⊈
38	String[] split(String,int)	r=str.split(v, i ₁)	r[i]	[v]{i}	-1	[^v]	i	⊈
		r=str.replace(v ₁ ,v ₂)		NULL	-1	[^v ₁]	0	⊈
	String replace(CharSequence,CharSequence)	r=str.replace(v ₁ ,v ₂)	NULL	NULL	-1	[^v ₁]	0	⊈
	String replaceAll(regex,String) String replaceFirst(String, String)	r=str.replaceAll(v ₁ ,v ₂)	NULL NULL	NULL NULL	-1 -1	$\begin{bmatrix} \mathbf{\hat{v}}_1 \end{bmatrix}$	0	⊈
-	String replaceFirst(String, String) String toLowerCase(Locale)	r=str.replaceFirst(v ₁ ,v ₂) r=str.toLowerCase(local)		NULL	-1 -1	[^v ₁] NULL	0	⊈
44	String toLowerCase()	r=str.toLowerCase()		NULL	-1	NULL	0	⊈
	String toUpperCase(Locale)	r=str.toUpperCase(local)	NULL	NULL	-1	NULL	0	⊈
	String toUpperCase()	r=str.toUpperCase()		NULL NULL	-1 1	NULL NULL	0	⊈
46	String to String	mate to blesse all	NULL	HNULL	-1	INULL		
46 47	String toString() String trim()	r=str.toString() r=str.trim()			-1	[^]	0	
46 47 48	String toString() String trim()	r=str.toString() r=str.trim() r=arr.length	NULL r==i	NULL [\s\S]{i}	-1 -1			<i>≢</i> ⊈ ≡
46 47 48 new		r=str.trim()	NULL	NULL	-1	[^]		⊈