Universitas Esa Unggul

Smart, Creative and Entrepreneurial

CMJ251-Manajemen Jaringan Mobile

www.esaunggul.ac.id

Dosen Pengampu :

5165-Kundang K Juman, Ir, MMSI

Prodi Teknik Informatika Fakultas Ilmu Komputer

Variabel

- Variabel bersifat case sensitive
- Teknik set nilai ke variabel pada WML:
  - Elemen <setvar>, yang dituliskan dalam elemen task <go>, <prev>, <refresh>
  - Elemen <input>
  - Elemen <select>

Pemberian variabel

- Variabel menyimpan data ketika seseorang user berpindah dari card satu ke card lain dalam sebuah deck
- Pemberian nilai ke sebuah variabel:

```
<setvar name="i" value="500"/>
```

Set nilai ke i

Setting variabel

```
<card id="card1">
   <select name="i">
   <option value="500">The Number 500</option>
   <option value="Five Hundred">500 in Text</option>
   </select>
</card>
```

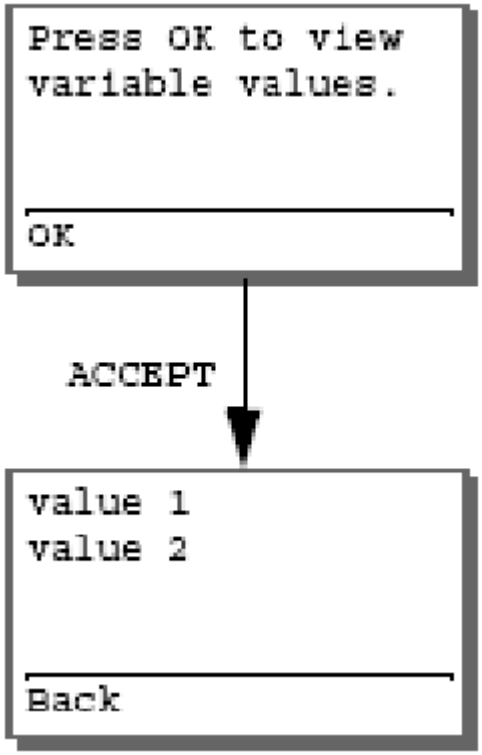
Set a value to i

Get the value of i

```
<card id="card2">
 <p>You selected: $(i)</p>
</card>
```

Contoh

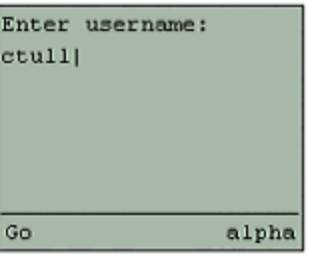
```
<wml>
  <card>
    <do type="accept">
      <go href="#viewvalues">
        <setvar name="var1" value="value 1"/>
        <setvar name="var2" value="value 2"/>
      </go>
    </do>
    <p>Press OK to view variable values.</p>
  </card>
  <card id="viewvalues">
    <p>$var1<br/>$var2</p>
  </card>
</wml>
```

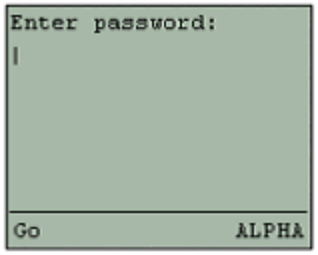


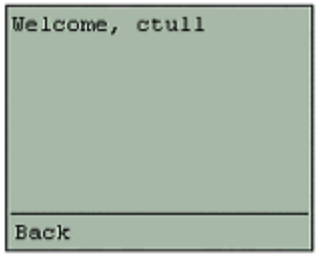
Form WML

```
<!-- form.wml -->
<wml>

  <!-- Card 1 -->
  <card id="username"
    title="Username">
    <do type="accept" label="Go">
      <go href="#password" />
    </do>
    <p>Enter username:
    <input name="username"
    type="text"/>
    </p>
  </card>
```

```
<!-- Card 2 -->
<card id="password" title="Password">
  <do type="accept" label="Go">
     <go href="#menu" />
  </do>
  <p>Enter password:
     <input name="password"
    type="password"/>
  </p>
</card>

<!-- Card 3 -->
<card id="menu" title="Menu">
   <p>Welcome, $(username)</p>
</card>

</wml>
```







Atribut <input> Format

Possible **format** attribute values	Description
A	Any uppercase letter or symbol; cannot be a digit.
a	Any lowercase letter or symbol; cannot be a digit.
M	Any character; uppercase characters take precedence.
m	Any character; lowercase characters take precedence.
N	Any numeric character or decimal point.
X	Uppercase characters only.
x	Lowercase characters only.
*Value	For instance, ***X** would allow the user to input any number of uppercase characters.
nValue	For instance, **5x** will accept exactly five lowercase characters

Atribut <input> lain

- emptyok
  - “true”, jika field boleh kosong
- maxlength
  - panjang maksimum field yang harus diisi
- size
  - lebar field masukan (dalam karakter)
- value
  - nilai awal field masukan
- name
  - nama objek <input>

Event onpick

- Tambahkan elemen <option> pada <select name=“bintang”> berikut
  - <option onpick="#pickother">Other sign ...</option>
- Lalu tambahkan card baru dengan id=“pickother”

```
<card id=“pickother">
  <do type="accept">
    <go href="#displaychoice">
      <setvar name=“bintang" value=“Bintang Anda: $bintang"/>
    </go>
  </do>
<p>Enter your sign:<input name=“bintang"/></p>
</card>
```

```xml
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
    <card id="satu" title="Satu">
        <do type="accept" label="Pilih Bintang">
            <go href="#display" />
        </do>
        <p>Pilih Bintang:<br/>
        <select name="bintang">
            <option value="capricorn">Capricorn</option>
            <option value="taurus">Taurus</option>
            <option value="sagitarius">Sagitarius</option>
            <option onpick="#pickother">Lainnya...</option>
        </select>
        </p>
    </card>
    <card id="pickother" title="Pilih Bintang Lain">
        <do type="accept" label="Submit">
            <go href="#display">
                <setvar name="choice" value="$bintang">
            </go>
        </do>
        <p>Masukkan bintangmu: <input name="bintang"/></p>
    </card>
    <card id="display">
        <p>Bintangmu adalah : <b>$bintang</b></p>
    </card>
</wml>
```

Hasil

Address file://E:\Documents\Dosen\mobile-computing\wml\coba31.wml

Pilih Bintang:

Capricorn

Pilih Bintang

Address file://E:\Documents\Dosen\mobile-computing\wml\coba31.wml#display

Bintangmu adalah : **taurus**

Address file://E:\Documents\Dosen\mobile-computing\wml\coba31.wml#pickother

Masukkan bintangmu: Aquarius

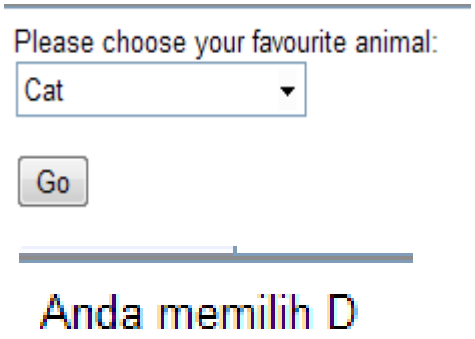
Submit

Address file://E:\Documents\Dosen\mobile-computing\wml\coba31.wml#display

Bintangmu adalah : **Aquarius**

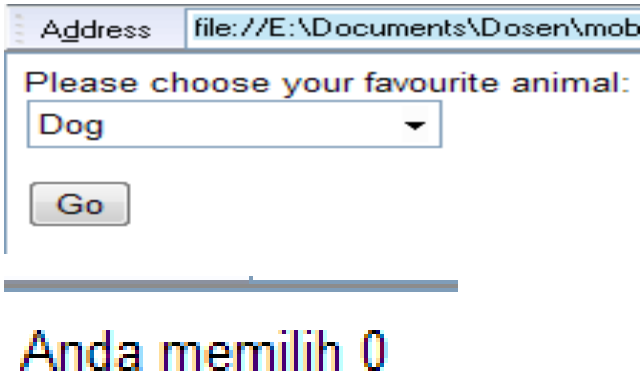
Select

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
 <card id="card1">
    <do type="accept" label="Go">
      <go href="#card2"/>
    </do>
    <p>Please choose your favourite animal:
    <select name="I">
        <option value="D">Dog</option>
        <option value="C">Cat</option>
    </select>
    </p>
  </card>

  <card id="card2">
    <p>Anda memilih $(I)</p>
  </card>
</wml>
```



Select (index)

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
 <card id="card1">
    <do type="accept" label="Go">
      <go href="#card2"/>
    </do>
    <p>Please choose your favourite animal:
    <select iname="I">
        <option value="D">Dog</option>
        <option value="C">Cat</option>
    </select>
    </p>
  </card>

  <card id="card2">
    <p>Anda memilih $(I)</p>
  </card>
</wml>
```



Fieldset

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
  <card>
    <p><do type="accept">
        <go
href="#submit"/>
    </do>
    <fieldset title="Name">First name:
      <input type="text" name="fname" maxlength="32"/><br/>
      Last name: <input type="text" name="lname" maxlength="32"/>
    </fieldset>
    <fieldset title="Info">
       <select name="sex">
           <option value="F">Female</option>
           <option value="M">Male</option>
       </select><br/>
       Age: <input type="text" name="age" format="*N"/>
    </fieldset>
    </p>
  </card>
  <card id="submit">
  <p>Ok, thx, $(fname), $(lname), $(sex), $(age).</p>
  </card>
</wml>
```

Optiongroup

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
    <card id="menu" title="Menu">
    <do type="accept" label="Go">
        <go href="#deptinfo"></go>
    </do>
    <p>Select a dept.:
    <select name="dept" value="5">
      <optgroup title="Departmen">
        <option value="Acct">Acct</option>
        <option value="Plant">Plant</option>
        <option value="IT">IT</option>
      </optgroup>
      <optgroup title="Title">
        <option value="Directur">Directur</option>
        <option value="Project Leader">Project Leader</option>
        <option value="Programmer">Programmer</option>
      </optgroup>
    </select>
    </p>
    </card>
    <card id="deptinfo" title="Dept Info">
    <p>This is the $(dept) dept.</p>
  </card>
</wml>
```

WMLScript

- WMLScript di dasarkan pada ECMAScript (ECMA262) yang menyediakan kemampuan scripting untuk WAP
- WMLScript merupakan bagian dari WML
- Dapat berjalan pada perangkat yang mendukung WAP
- Case sensitive
- WMLScript mengabaikan spasi, tab atau ganti baris antar token, kecuali pada sebuah string
- Komentar pada WMLScript:
  - Baris: //
  - Blok: /* */

WMLScript

- WMLScript is a light version of the JavaScript language
- WML scripts are not embedded in the WML pages. WML pages only contains references to script URLs
- WMLScript is compiled into byte code on the server before it is sent to the WAP browser
- WMLScript is a part of the WAP specification

- Kegunaan:
  - WMLScript is used to validate user input
  - WMLScript is used to generate message boxes and dialog boxes locally, to view error messages and confirmations faster

Literal

- Literal adalah nilai yang ditulis pada program script
- Ada 5 literal:
  - Integer literal
  - Floating Point literal
  - String literal
  - Boolean literal
  - Invalid literal

Integer dan Floating Point

- Integer
  - Desimal (base 10), contoh: 32
  - Hexadesimal (base 16), contoh: 0x20
  - Oktal (base 8), contoh: 040
- Floating point
  - Minimal 1 digit dan baik titik desimal atau eksponen
  - Eksponen: nilai 10 logaritma. contoh: e0 = 10^0
  - Contoh: 3.14, 0.314e1, 314e-2

String Literal

- Sembarang urutan kosong atau lebih karakter yang diapit oleh ‘ atau “

Character represented	Symbol	Sequence
Apostrophe or single quote	'	\'
Double quote	"	\"
Backslash	\	\\
Slash	/	\/
Backspace		\b
Form feed		\f
Newline		\n
Carriage return		\r
Horizontal tab		\t

Boolean Literal

- Boolean literal: true atau false
- Invalid literal: variabel yang menandakan nilai tidak sah (invalid)
- Contoh invalid:

```
var x = 8;
var y = 0;
if ((x/y) == invalid) {
    //display error message
}
```

Deklarasi Variabel

- Deklarasi
  - Menyatakan nama variabel
  - Variabel harus dideklarasikan dahulu sebelum digunakan
  - Contoh:
    - var x, y;
    - var ukuran = “M”;
- Variabel WMLScript hanya dapat diakses dalam fungsi yang mendeklarasikan
- Nama variabel dalam satu fungsi harus unik

Setvar

- Gunakan fungsi setvar dari class WMLBrowser
- WMLBrowser.setVar("greeting", “Met Datang\nWMLScript!");
- Dengan fungsi tersebut, variabel $greeting dapat diakses dari dokumen WML

WMLScript

Metdatang.wmls

```
extern function metdatang() {
    WMLBrowser.setVar( "teks","Met Datang di Pemrograman WMLScript!" );
    WMLBrowser.refresh();
}
```

hallo.wml

```
<wml>
   <card id = "index" title = "Line" newcontext = "true">
      <onevent type = "onenterforward">
         <go href = "metdatang.wmls#metdatang()" />
      </onevent>
      <p>$teks</p>
   </card>
</wml>
```

Reserved words

Reserved Identifiers					
access	debugger	false	isvalid	sizeof	use
agent	default	finally	lib	struct	user
break	do	for	meta	super	var
case	domain	function	name	switch	void
catch	else	header	new	this	while
class	enum	http	null	throw	with
const	equiv	if	path	true	
continue	export	import	private	try	
delete	extends	in	public	typeof	
div	extern	invalid	return	url	

Operator

Operator	Operation
+	add (numbers)/concatenation (strings)
-	subtract
*	multiply
/	divide
div	integer division

Operator	Operation
+	plus
-	minus
--	pre-or-post decrement
++	pre-or-post increment
~	bitwise NOT

Operator	Operation
%	remainder or modulus, the sign of the result equals the sign of the dividend
<<	bitwise left shift
>>	bitwise right shift
>>>	bitwise shift right with zero fill
&	bitwise AND
|	bitwise OR
^	bitwise XOR

Operator Logika

- Jika operan pertama untuk logika AND atau OR adalah invalid invalid, maka operan kedua tidak dicek dan menghasilkan invalid

Operator	Operations
&&	logical AND
||	logical OR
!	logical NOT (unary)

Operator logika

```
extern function metdatang(){
  WMLBrowser.setVar("teks","Hallo");
  WMLBrowser.refresh();
}

extern function cobavariabel(){
  var a = true || false;
  var b = false && false;
  var c = true && false;

  WMLBrowser.setVar("aa",a);
  WMLBrowser.setVar("bb",b);
  WMLBrowser.setVar("cc",c);
  WMLBrowser.refresh();
}
```

Operator String

- Untuk menggabungkan dua string dapat digunakan + atau +=
- Operator perbandingan

Operator	Operation
<	less than
<=	less than or equal
==	equal
>=	greater than or equal
>	greater than
!=	inequality

Statement

- Empty
  while (true);
- Expression
  myValue1 = counter, myValue 2 = val13;
- Block
  { //begin block statement
    var i = 0;
    var x = Lang.abs(b);
    popUp("Remember!");
  } //end block statement

```
function count(str) {
  var result = 0; // Initialized once
  while(str != "") {
    var ind = 0; // Initialized every time
    // modify string
  }
  return result
}
```

Obyek WMLBrowser

Method	Description	Example
`getCurrentCard`	returns the URL of the card being displayed.	`getCurrentCard()`
`getVar`	returns the value of a browser variable.	`getVar( "var1" )`
`go`	sends a user to a given location.	`go( "#card2" )` redirects to `card2`
`newContext`	clears all variable values.	`newContext()`
`prev`	returns to the previous card.	`prev()`
`refresh`	refreshes the display of the WML card	`.refresh()`
`setVar`	sets the value of a browser variable.	`setVar( "x", "Hello" )` gives the value `"Hello"` to `x`

Getvar.wml

```
<wml>
  <card id = "index" title = "getVar">
    <do type = "accept" label = "Run">
      <go href =
"getvar.wmls#getName()" />
    </do>
    <p>Enter your name:<br />
    <input name = "name" value = "" />
    </p>
  </card>
  <card id = "card2" title = "getVar">
    <do type = "accept" label = "Back">
      <prev />
    </do>
    <p>Your name is:<br />$result</p>
  </card>
</wml>
```

Getvar.wmls

```
extern function getName(){
  var x = WMLBrowser.getVar(
"name" );
  var y = x + "\nThanks for visiting!";
  WMLBrowser.setVar( "result", y );
  WMLBrowser.go( "#card2" );
}
```

Obyek Float

Method	Description	Example
`round( x )`	rounds *x* to the closest integer	`round( 9.75 )` is `10` `round( 9.25 )` is `9`
`ceil( x )`	rounds *x* to the smallest integer not less than *x*	`ceil( 9.2 )` is `10.0` `ceil( -9.8 )` is `-9.0`
`floor( x )`	rounds *x* to the largest integer not greater than *x*	`floor( 9.2 )` is `9.0` `floor( -9.8 )` is `-10.0`
`int( x )`	truncate a floating point value into its integer form.	`int( 2.718282 )` is `2` `int( 7.389056 )` is `7`
`maxFloat()`	the largest floating-point value that can be represented	`maxfloat()` is `3.4028235E38`
`minFloat()`	the smallest floating-point value that can be represented	`minFloat()` is `1.17549435E-38`
`pow( x, y )`	*x* raised to power *y* (xy)	`pow( 2.0, 7.0 )` is `128.0` `pow( 9.0, .5 )` is `3.0`
`sqrt( x )`	square root of *x*	`sqrt( 900.0 )` is `30.0` `sqrt( 9.0 )` is `3.0`

Percobaan Float

```
<wml>
  <card id = "index" title = "Float">
    <do type = "accept" label = "Run">
      <go href = "tryFloat.wmls#numbers( $input )" />
    </do>
    <p>Enter a floating-point number:<br />
    <input name = "input" value = "" /></p>
  </card>
  <card id = "card2" title = "Float">
    <do type = "accept" label = "Home">
      <prev />
    </do>
    <p>round( $input ) = $roundResult <br />
    ceil( $input ) = $ceilResult <br />
    floor( $input ) = $floorResult <br />
    int( $input ) = $intResult</p>
  </card>
</wml>
```

tryFloat.wmls

```
extern function numbers( userInput ){
  var number = Lang.parseFloat( userInput );

  WMLBrowser.setVar( "roundResult", Float.round( userInput ) );
  WMLBrowser.setVar( "ceilResult", Float.ceil( number ) );
  WMLBrowser.setVar( "floorResult", Float.floor( number ) );
  WMLBrowser.setVar( "intResult", Float.int( number ) );
  WMLBrowser.go( "#card2" );
}
```

Obyek Lang

Method	Description	Example
abort	stops the execution of the script and returns control to the browser.	**abort(** "Operatopm Failed" **)**
abs	calculates the absolute value of a number.	**abs(** -21 **)** is 21
exit	stops execution of the script and returns control to script interpreter.	**exit(** 5 **)** stops script
float	returns a boolean if the browser supports floating-point numbers.	**minFloat()** is **true** or **false**
isFloat	returns a boolean if a number is a floating-point number.	**isFloat(** "8.754" **)** is **true**
isInt	returns a boolean if a number is an integer.	**isInt(** "98" **)** is **true**
max	calculates the greater of **x** and **y**.	**max(** 5, 4 **)** is 5
maxInt	calculates the largest integer value supported by the browser.	**maxInt()** is 2147483647
min	calculates the smaller of **x** and **y**.	**min(** 5, 4 **)** is 4
minInt	calculates the smallest integer value supported by the browser.	**minInt()** is -2147483648
parseInt	converts the value to an integer	**parseInt(** "7" **)** is 7
parseFloat	converts the value to a floating-point number.	**parseFloat(** "3.3178" **)** is 3.3178
random	returns a random integer between 0 and **x**	**round(** 3 **)** is 0, 1, 2 or 3
seed	specifies a seed from the random function. Specifying a new seed each time random is called will help ensure the uniqueness of each random number.	**Lang.seed(** 34 **)** seeds the random function with 34.

Contoh Lang

```
<wml>
  <card id = "index" title = "Lang">
    <do type = "accept" label = "Run">
      <go href = "tryLang.wmls#langFunctions( $input )" />
    </do>
    <p>Enter a number<br />
    <input name = "input" value = "" /></p>
  </card>
  <card id = "card2" title = "Lang">
    <do type = "accept" label = "Kmbl">
      <prev/>
    </do>
    <p>Mendukung float?<br />$result1</p>
    <p>Nilai absolut $input :<br />$result2</p>
    <p>Dapatkah $result2 diparsing sebagai float?<br />$result3</p>
    <p>Dapatkah $result2 diparsing sebagai integer?<br />$result4</p>
  </card>
</wml>
```

TryLang.wmls

```
extern function langFunctions( userInput ){
  var variable1 = userInput;

  // check if browser handles floats
  var variable2 = Lang.float();
  var variable3 = String.format( "%3.2f", Lang.abs( variable1 ) );
  var variable4 = Lang.isFloat( variable3 );
  var variable5 = Lang.isInt( variable3 );

  WMLBrowser.setVar( "result1", variable2 );
  WMLBrowser.setVar( "result2", variable3 );
  WMLBrowser.setVar( "result3", variable4 );
  WMLBrowser.setVar( "result4", variable5 );
  WMLBrowser.go( "#card2" );
}
```

tryString.wml

```
<wml>
  <card id="card1" title="Masukan String" newcontext="true">
    <do type="accept" label="Proses">
      <go href="tryString.wmls#tryStr()" />
    </do>
    <p>String 1: <input name="str1" value=""/><br/>
      String 2: <input name="str2" value=""/><br/>
    </p>
  </card>

  <card id="card2" title="Hasil">
    <p><b>Panjang str1: $panjang1</b></p>
    <p><b>Panjang str2: $panjang2</b></p>
    <p><b>Perbandingan str1 dan str2: $compare</b></p>
    <p><b>Lima karakter pertama str1: $substr</b></p>
    <p><b>Squeeze str1: $squeeze</b></p>
    <p><b>Apakah str2 kosong?: $empty</b></p>
  </card>
</wml>
```

String

Method	Description
charAt(*index* **)**	Returns the character at the specified *index*. If there is no character at that *index*, **charAt** returns an empty string. The first character is located at *index* 0.
compare(*string1, string2* **)**	Compares the values of *string1* and *string2* and returns an integer value of **-1**, **0**, or **1**. A value of **-1** is returned if *string1* is less than *string2*. A value of **0** denotes that the two strings are equal. A value of **1** is returned if *string1* is greater than *string2*.
elementAt(*string, index, separator* **)**	Searches string starting with the indexth element separated by separator.
elements(*string, separator* **)**	Returns the number of elements contained in string separated by separator.
find(*string, substring* **)**	Returns the index value of the first occurrence of substring in string.
format(*format, value* **)**	Returns value in the format specified in format. The format value begins with a percent sign (%) followed by the width argument, the precision arguments and the type argument. The width argument defines the minimum number of characters to the left of the decimal point. The precision arguments defines a number preceded by a decimal and is defined by the type argument. The value of the type argument can be either f, d or s. A value of defines the minimum number of characters to the left of the decimal point. A value of d defines the minimum number of characters to be printed. A value of s defines the maximum number of characters to be printed.
insertAt(*string, element, index, separator* **)**	Returns a new string with the element and the separator inserted at position index.
isEmpty(*string* **)**	Returns a boolean value of true or false. If the length of the string is zero, a boolean true is returned. Otherwise, a boolean false is returned.

String (2)

Method	Description
`length(` *string* **`)`**	Returns the length of the string given.
`removeAt(` *string*, *index*, *separator* **`)`**	Removes the contents of the string starting at the index position and ending at the specified separator.
`replace(` *string*, *old*, *new* **`)`**	Creates a new string containing the contents of *new* in place of *old* in *string*.
`replaceAt(` *string*, *element*, *index*, *separator* **`)`**	Creates a new string replacing the existing element starting at position index with the new element ending at the specified separator.
`squeeze(` *string* **`)`**	Returns the given string replacing any series of whitespaces with a single space.
`substr(` ***string*, *startIndex*, *length*** **`)`**	Returns a string containing *length* characters starting from *startIndex* in the source string. If *length* is not specified, a string containing characters from *start* to the end of the source string is returned.
`toString(` *value* **`)`**	Returns the given value as type string.
`trim(` *string* **`)`**	Returns the give string minus any whitespace at the beginning or end of the string.

Format String

```
var a = 45;
var b = -45;
var c = "now";
var d = 1.2345678
var e = String.format("e: %6d", a); // e = "e: 45"
var f = String.format("%6d", b); // f = " -45"
var g = String.format("%6.4d", a); // g = " 0045"
var h = String.format("%6.4d", b); // h = " -0045"
var i = String.format("Do it %s", c); // i = "Do it now"
var j = String.format("%3f", d); // j = "1.234567"
var k = String.format("%10.2f%%", d); // k = " 1.23%"
var l = String.format("%3f %2f.", d); // l = "1.234567."
var m = String.format("%.0d", 0); // m = ""
var n = String.format("%7d", "Int"); // n = invalid
var o = String.format("%s", true); // o = "true"
```

tryString.wml

```
<wml>
  <card id="card1" title="Masukan String" newcontext="true">
    <do type="accept" label="Proses">
      <go href="tryString.wmls#tryStr()" />
    </do>
    <p>String 1: <input name="str1" value=""/><br/>
      String 2: <input name="str2" value=""/><br/>
    </p>
  </card>

  <card id="card2" title="Hasil">
    <p><b>Panjang str1: $panjang1</b></p>
    <p><b>Panjang str2: $panjang2</b></p>
    <p><b>Perbandingan str1 dan str2: $compare</b></p>
    <p><b>Lima karakter pertama str1: $substr</b></p>
    <p><b>Squeeze str1: $squeeze</b></p>
    <p><b>Apakah str2 kosong?: $empty</b></p>
  </card>
</wml>
```

tryString.wmls

```
extern function tryStr() {
  var param1 = String.toString(WMLBrowser.getVar("str1"));
  var param2 = String.toString(WMLBrowser.getVar("str2"));

  WMLBrowser.setVar("panjang1", String.length(param1));
  WMLBrowser.setVar("panjang2", String.length(param2));
  WMLBrowser.setVar("compare", String.compare(param1,
param2));
  WMLBrowser.setVar("substr", String.subString(param1, 0, 5));
  WMLBrowser.setVar("squeeze", String.squeeze(param1));
  WMLBrowser.setVar("empty", String.isEmpty(param2));
  WMLBrowser.go( "#card2" );
}
```

Obyek Dialogs

Method	Description

alert(*string* **)**	Menampilkan pesan ke user, menunggu konfirmasi dari user, dan mengembalikan ""
	Contoh:
	`if (String.length(textElement) > 8) {`
	`Dialogs.alert("Enter name < 8 chars!");`
	`};`

confirm(*message*, *ok*, *cancel***)**	Menampilkan pesan ke user dengan dua pilihan: ok dan cancel. Akan mengembalikan true (ok) dan false (cancel)
	Contoh:
	`function onAbort() {`
	`return Dialogs.confirm("Are you sure?","Well...","Yes");`
	`};`

prompt(*message*, *defaultInput***)**	Menampilkan pesan dan prompt untuk menerima masukan dari user. Mengembalikan masukan user.
	Contoh:
	`var a = "234-1234";`
	`var b = Dialogs.prompt("Phone number: ",a);`

Pemanggilan fungsi

- Ada 3 pemanggilan fungsi:
  - Local script function
  - External function
  - Library function
- Agar suatu fungsi dapat dipanggil/direferensikan dari suatu WML, maka fungsi tersebut harus dinyatakan sebagai extern.

Load Script Function

- Suatu fungsi yang dideklarasikan dan dipanggil pada satu file yang sama
- Contoh:

```
function test2 (param) {
  return test1 (param+1);
};
function test1 (val) {
  return val*val;
};
```

Fungsi external

- Suatu fungsi yang dideklarasikan pada file terpisah (eksternal)
- Definisi lokasi fungsi menggunakan pragma **use url**.

```
use url OtherScript
  "http://www.host.com/script";
function test3(param) {
  return OtherScript#test2(param+1);
};
```

- Suatu fungsi yang memanggil fungsi pustaka standard WMLScript

```
function test4(param) {
  return Float.sqrt(Lang.abs(param)+1);
};
```

Statemen IF

```
var pn = String.length(param1);
 if (pn >= 20)
    WMLBrowser.setVar("kat", "A");
 else if (pn >= 10)
    WMLBrowser.setVar("kat", "B");
 else
    WMLBrowser.setVar("kat", "C");
```

Statemen Perulangan (while-for)

```
var counter = 0;
var total = 0;
while (counter < 3) {
  counter++;
  total += c;
};
```

```
for (var index = 0; index < 100; index++) {
  count += index;
  myFunc(count);
};
```

Statement Break, Continue, dan Return

```
function testBreak(x) {
  var index = 0;
  while (index < 6) {
     if (index == 3) break;
       index ++
  };
  return index*x;
};
```

```
var index = 0
var count = 0
while (index < 5) {
  index++;
  if (index == 3) continue;
  count += index;
};
```

```
function square (x) {
  if (!(Lang.isFloat(x))) return invalid;
  return x * x;
}
```