QT4 CG★ Status Update

Assembled by Christian Grün | BaseX

★X(Q)uery and XSL(T) 4 Community Group

QT4 » History

2017	XPath & XQuery 3.1, XSLT 3.0	
2017, Oct	QTSpecs moved from CVS to GitHub https://github.com/qt4cg/qtspecs	...thanks Liam!
2018, Oct	XPath Syntax Extensions Wishlist https://github.com/expath/xpath-ng	...thanks Adam!
2020, Apr	XSLT Extensions Community Group https://w3.org/community/xslt-40/	...thanks Mike!
2020, Jun	QT4 CG Home Page https://qt4cg.org	...thanks Norm!
2022	Regular QT4 CG meetings	...thanks all!

QT4 » Regular Participants

Chair, Scribe	Norm Tovey-Walsh
Co-Chair	C. M. Sperberg-McQueen
Members★	Dimitre Novatchev Joel Kalvesmaki Christian Grün Sasha Firsov John Lumley Michael Kay Reece Dunn Ed Porter

★order by fn:string-length descending

QT4 » Objectives

- Incorporate user feedback
- Let 3.0/3.1 features further evolve
- Include functionality missing in daily use
- Embed and unify common vendor-specific extensions
- Integrate popular features from other programming languages
- Keep the languages alive!

QT4 » What you will get

Otherwise operator

```
$a otherwise $b
```

...equivalent to...

```
if(exists($a)) then $a else $b
```

Braced if, optional else

```
if($a) { $b } else { $c }
if($a) { if($b) { $c } }
```

Arrow map operator

```
"The cat sat on the mat"
=>  tokenize()
=!> concat(".")
=!> upper-case()
=>  string-join(" ")
```

Numeric literals

```
1_048_576,
0xFC00,
0b11110000
```

QT4 » What you will get

Abbreviated function syntax

```
let $inc := fn($n) { $n + 1 }
return $inc(99)
```

...equivalent to...

```
let $inc := function($n) {
  $n + 1
}
return $inc(99)
```

Focus functions (arity-one)

```
let $inc := fn { . + 1 }
return $inc(99)

fn:filter($integers, fn { . > 5 })

fn:iterate-while(
  $number,
  fn { . < 100 },
  fn { . * . }
)
```

QT4 » What you will get

Optional arguments

```
declare function strings:get(
  $value,
  $default := "EMPTY"
) {
  $value otherwise $default
}
strings:get('john')
```

Keyword arguments

```
sort($cars, key := fn { @age })

declare function coordinates(
  $x := 0,
  $y := 0
) {
  map { "x": $x, "y": $y }
}
coordinates(y := 123)
```

QT4 » What you will get

String templates

```
`Name: { $name }, { $age }`
```

...equivalent to...

```
"Name: " || $name ||
  ", " || $age
```

...or...

```
``[Name: `{ $name }`, { $age }]``
```

Union node tests

```
/xml/child::(a|b),
/xml/element(c|d)
```

Compact lookup syntax

```
$address?$name,
$city?"city code"
```

...equivalent to...

```
$address?($name),
$city?("city code")
```

QT4 » What you will get

Functions, functions, functions★

fn:all-different, fn:all-equal, fn:build-uri, fn:char, fn:characters, fn:contains-sequence, fn:duplicate-values, fn:every, fn:foot, fn:highest, fn:identity, fn:index-where, fn:is-NaN, fn:items-after, fn:items-before, fn:iterate-while, fn:log, fn:lowest, fn:op, fn:parse-integer, fn:parse-uri, fn:partition, fn:replicate, fn:slice, fn:some, fn:transitive-closure, fn:trunk, fn:void

array:build, array:foot, array:members, array:of-members, array:slice, array:split, array:trunk

map:build, map:filter, map:of-pairs, map:pair

★looking for more functions? https://qt4cg.org/specifications/xpath-functions-40

QT4 » What may you get?

Context item ⟶ value

Bind sequences to context:

```
declare context value :=
  collection('books');
//title

array:sort(
  $assets,
  key := fn { count(.) }
)
```

FLWOR extensions

```
for member $m in $array
return count($m)

for key $k value $v in $map
return $k || ': ' || $v

for $n in $integers
while $n < 10
return $n
```

QT4 » What may you get?

Destructuring assignments

Sequences

```
let $(sin,cos) := sincos(...)
```

Arrays

```
let $[first, others] := array {...}
```

Maps

```
let ${x, y} := map {...}
```

Variadic parameters

```
fn($arg... as xs:string*) { ... }
```

To be continued...

- maps&arrays: navigation
  `$book?chapter[.??line = ...]`
- maps&arrays: updates
- new data structure: sets?

...what are you missing?

Thanks. Everyone. Participate!