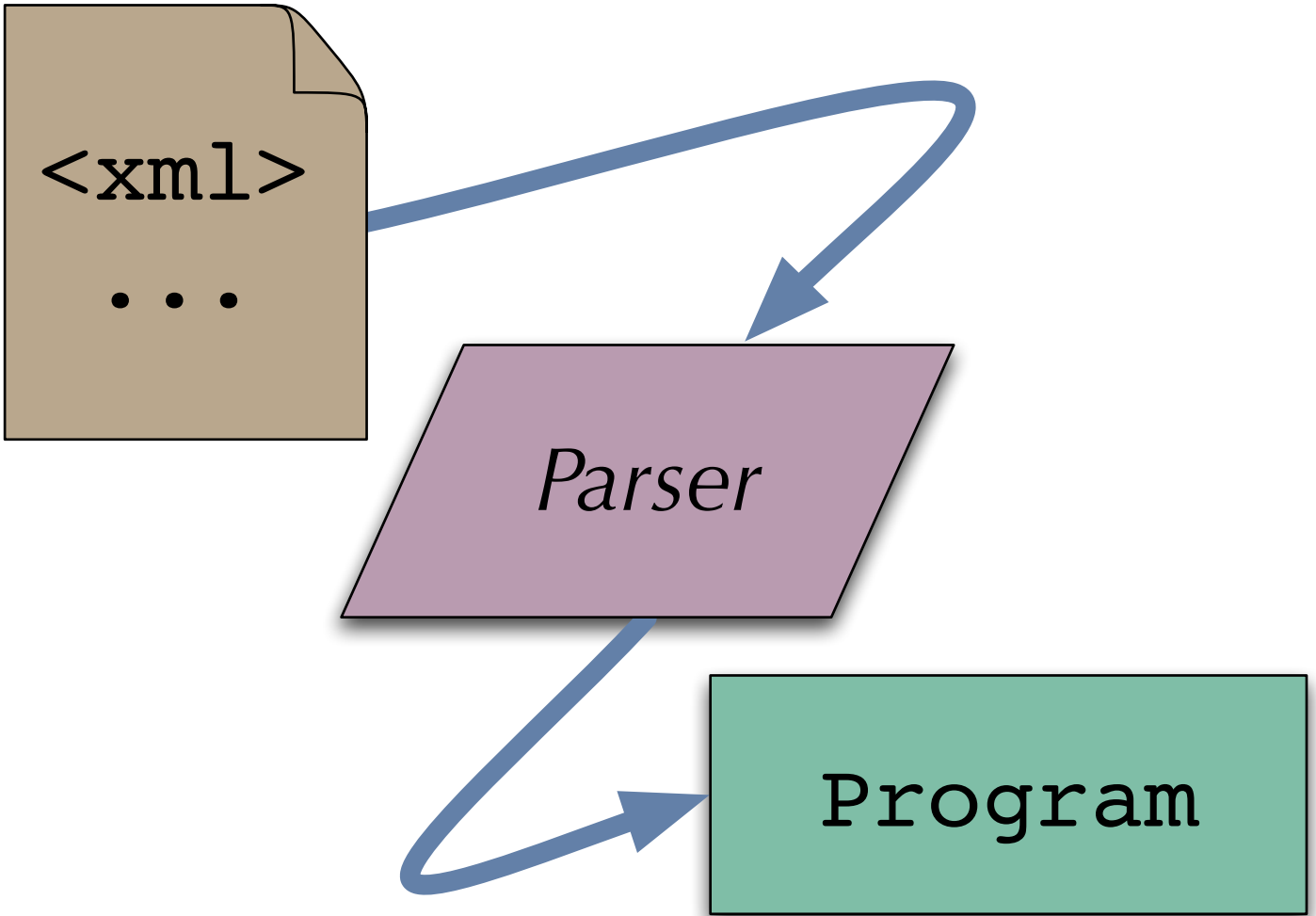
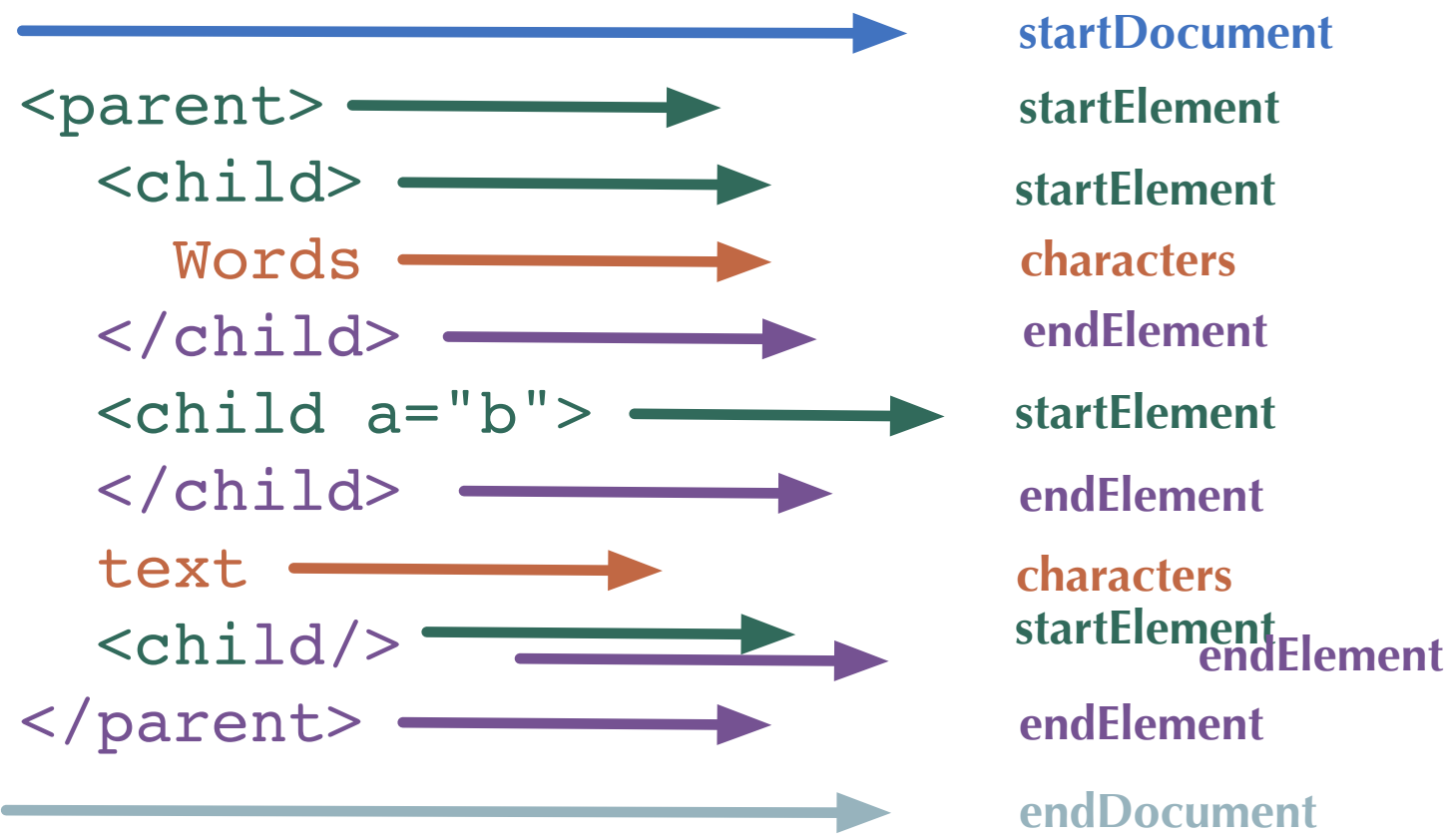
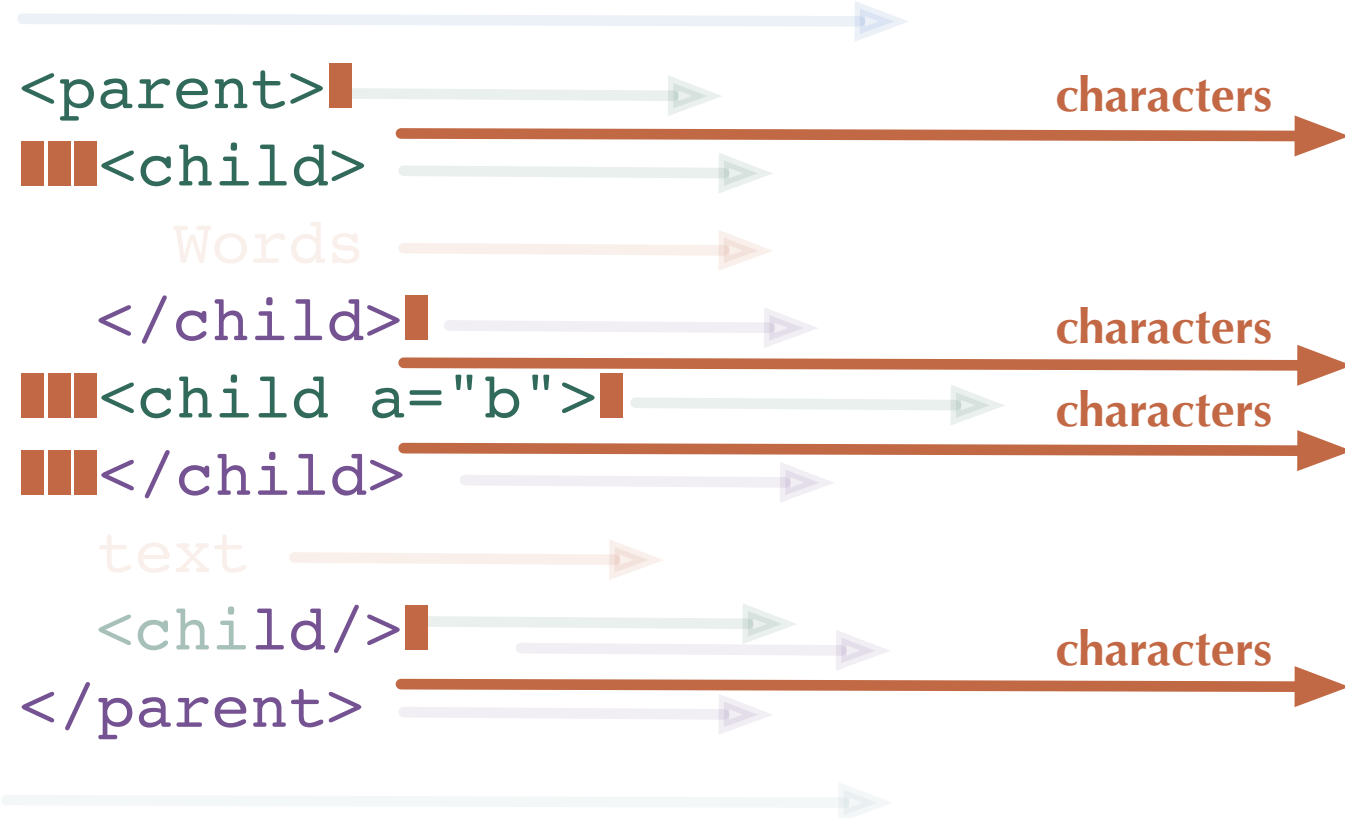


Introduction to SAX



```
<parent>
  <child>
    Words
  </child>
  <child a="b">
  </child>
  text
  <child/>
</parent>
```





`<parent>`

`<child>`

Words

`</child>`

`<child a="b">`

`</child>`

text

`<child/>`

`</parent>`

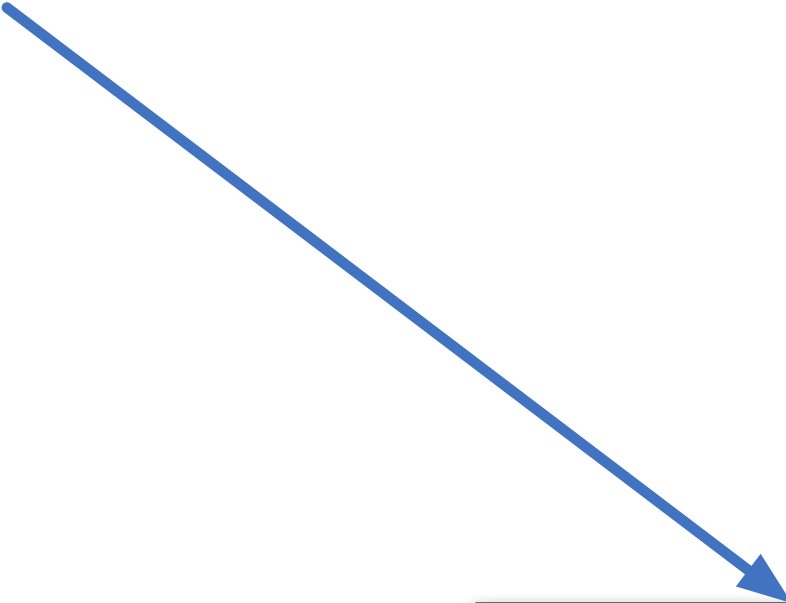
characters

characters

characters

characters

startDocument



Program

startElement



name

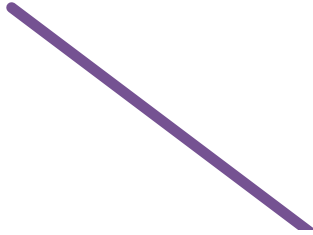


**attribute
list**

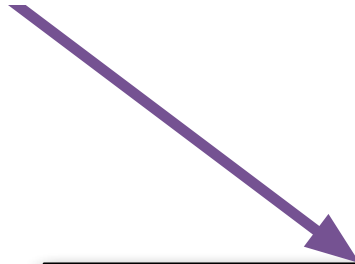


Program

endElement

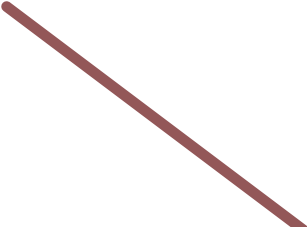


name

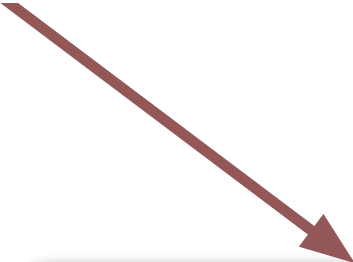


Program

characters

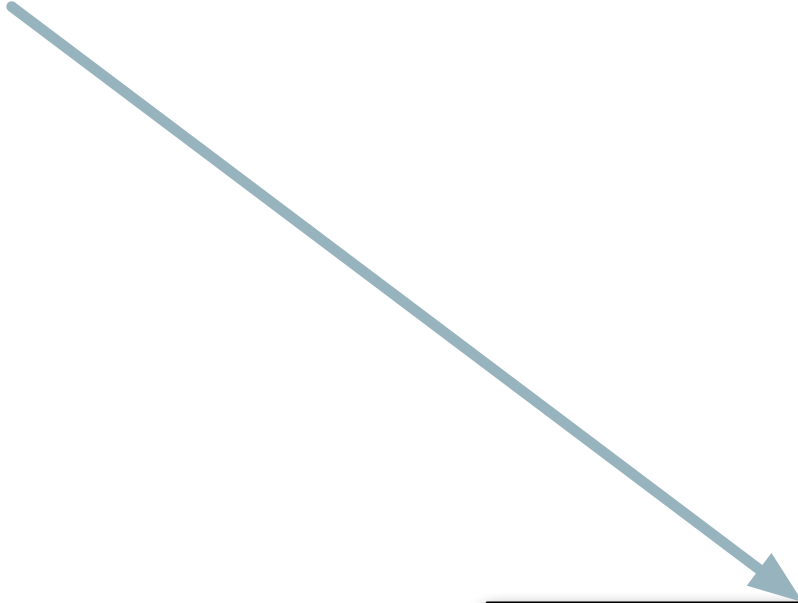


chars



Program

`endDocument`



Program

startDocument

startElement

name

**attribute
list**

endElement

name

characters

chars

endDocument

```
program science
```

```
  use FoX_sax
```

```
  ...
```

```
module science
```

```
  use FoX_sax
```

```
  ...
```

```
subroutine science
```

```
  use FoX_sax
```

```
  ...
```

```
function science
```

```
  use FoX_sax
```

```
  ...
```

File descriptor

```
program science
  use FoX_sax
  type(xml_t) :: xp

  ...

  call xml_doStuff(xp, ...)
```

```
type(xml_t) :: xp
```

```
call open_xml_file(xp, fileName, unit)
```

```
call parse(xp, ...)
```

```
call close_xml_t(xp)
```

Event handler

```
subroutine hello()  
    print*, "hello, someone called?"  
end subroutine
```

```
type(xml_t) :: xp
```

```
call open_xml_file(xp, fileName, unit)
```

```
call parse(xp, &  
    startDocument_handler = hello)
```

```
call close_xml_t(xp)
```

Event handler with data

```
subroutine print_string(chars)
  character(len=*), intent(in) :: chars

  print*, "Received some chars:"
  print*, chars
end subroutine
```

```
type(xml_t) :: xp
```

```
call open_xml_file(xp, fileName, unit)
```

```
call parse(xp, &  
    characters_handler = print_chars)
```

```
call close_xml_t(xp)
```

```
type(xml_t) :: xp

call open_xml_file(xp, fileName, unit)

call parse(xp,                                     &
           startDocument_handler = hello,         &
           characters_handler = print_chars,      &
           endDocument_handler = hello            )

call close_xml_t(xp)
```

Attributes

name = "value"

1: day = "Tuesday"

2: date = "9"

3: month = "January"

4: year = "2007"

```
type(dictionary_t) :: dict
```

```
i = len(dict)
```

```
print*, getQName(dict, 3)
```

```
print*, getValue(dict, 3)
```

```
type(dictionary_t) :: dict
```

```
print*, hasKey(dict, "day")
```

```
print*, getValue(dict, "month")
```

startElement event handler

```
subroutine se_handler(name, dict)
  use FoX_sax
  character(len=*), intent(in) :: name
  type(dictionary_t), intent(in) :: dict
  integer :: i

  do i = 1, len(dict)
    print*, getQName(dict,i)
    print*, getValue(dict,i)
  enddo
end subroutine
```

```
type(xml_t) :: xp

call open_xml_file(xp, fileName, unit)

call parse(xp,                                     &
           startDocument_handler = hello,         &
           startElement_handler = se_handler, &
           characters_handler = print_chars,     &
           endElement_handler = print_chars, &
           endDocument_handler = hello           )

call close_xml_t(xp)
```

Namespaces

```
subroutine se_handler_ns(nsURI, localName, &  
                        QName, dict)  
  
  use FoX_sax  
  character(len=*), intent(in) :: nsURI  
  character(len=*), intent(in) :: localName  
  character(len=*), intent(in) :: QName  
  type(dictionary_t), intent(in) :: dict  
  
  ...  
  
end subroutine
```

Namespaces

```
subroutine ee_handler_ns(nsURI, localName, &  
                        QName)  
  
  character(len=*), intent(in) :: nsURI  
  character(len=*), intent(in) :: localName  
  character(len=*), intent(in) :: Qname  
  
  ...  
  
end subroutine
```

Storing data

- module/common variables
- call out to other subroutine

