

 免费电子书

学习

xpath

Free unaffiliated eBook created from
Stack Overflow contributors.

#xpath

.....	1
1: xpath	2
.....	2
.....	2
Examples.....	2
XML.....	2
.....	2
.....	3
HTML.....	3
XPath.....	3
2:	5
.....	5
Examples.....	5
.....	5
.....	5
.....	5
“”.....	6
.....	6
.....	6
3:	7
.....	7
Examples.....	7
.....	7
4:	8
.....	8
.....	8
.....	8
Examples.....	8
.....	8
.....	8
.....	9
.....	9

.....	10
.....	10
5:	11
Examples	11
.....	11
.....	11
6:	12
Examples	12
.....	12
7:	14
Examples	14
.....	14
.....	14
8:	16
.....	16
.....	16
Examples	16
.....	16
.....	16
.....	16
9:	18
.....	18
.....	18
Examples	18
Deborah	18
Dobby	18
10:	20
.....	20
.....	20
.....	20

Examples.....	20
.....	20
.....	21
.....	21
.....	21
Parashurama.....	22
Parashurama.....	22
Parusharama.....	23
House.....	24
Room.....	24
.....	24
11:	26
.....	26
.....	26
.....	26
Examples.....	26
.....	26
.....	26
12:	27
.....	27
.....	27
.....	27
Examples.....	27
LightDeviceSensor.....	27
Light.....	27
Star.....	28
Ball.....	29
light.....	30
.....	31

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [xpath](#)

It is an unofficial and free xpath ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official xpath.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: xpath

XPathXML。

XSLTXQuery。。

XPathW3C

- XPath 1.0 [XMLXPath1.0](#)
- XPath 2.0 [XMLXPath2.0](#)
- XPath 3.0 [XMLXPath3.0](#)

1.0	1999-12-16
2.0	2007-01-23
3.0	201448
3.1 W3C	20151217

Examples

XML

XMLXPath

```
<r>
  <e a="1"/>
  <f a="2" b="1">Text 1</f>
</f>
<g>
  <i c="2">Text 2</i>
  Text 3
  <j>Text 4</j>
</g>
</r>
```

XML

XPath

```
/r/f/text()
```

```
"Text 1"
```

XPath

```
string(/r/f)
```

f

```
"Text 1"
```

XML

XPath

```
/r/e
```

```
<e a="1"/>
```

HTML

HTML DOM

```
<html>
  <body>
    <a>link</a>
    <div class='container' id='divone'>
      <p class='common' id='enclosedone'>Element One</p>
      <p class='common' id='enclosedtwo'>Element Two</p>
    </div>
  </body>
</html>
```

ID

```
//*[@id='divone'] # Returns <div class='container' id='divone'>
```

ID

```
/html/body/div/p[@id='enclosedone'] # Returns <p class='common' id='enclosedone'>Element One</p>
```

id

```
//p[@id='enclosedone' and @class='common'] # Returns <p class='common' id='enclosedone'>Element One</p>
```

```
//*[@id='enclosedone']/text() # Returns Element One
```

Xpath

xpath

```
$x('//insert xpath here')
```

\$ - °

x - xpath

```
$x("//button[text()='Submit']")
```

Submit°

xpath [https://riptutorial.com/zh-CN\(xpath/topic/883\(xpath](https://riptutorial.com/zh-CN(xpath/topic/883(xpath)

2:

XPath /

```
step1/step2/step3
```

◦ ::◦

```
axis::nodeTest [predicate1] [predicate2]
```

/◦ *N*

- *N*
-
- ◦

◦

Examples

child

```
/child::html/child::body/child::div/child::span
```

child

```
/html/body/div/span
```

descendant◦ descendant-or-self◦ child◦

```
/child::html/descendant::span  
/child::html/descendant-or-self::*
```

///
descendant-or-self::node() /◦

```
table//td  
child::table/descendant-or-self::node()/child::td  
child::table/descendant::td  
table/descendant::td
```

parent◦ htmlbody

```
/child::html/child::body/parent::html
```

..parent::node()

ancestor◦ ancestor-or-self◦ div

```
ancestor::div
```

“”

```
self◦ .self::node()◦ .◦
```

```
./span  
self::node()/descendant-or-self::node()/child::span  
descendant::span
```

```
selfXPath 1.0◦ h1 h2h3
```

```
*[self::h1 or self::h2 or self::h3]
```

following-siblingpreceding-siblingfollowingpreceding

- ◦
- following◦
- preceding◦

```
following::span[1]  
following-sibling::*[last()]
```

```
attributnamespace◦ @attribute::
```

```
child::div/attribute::class  
div/@class
```

<https://riptutorial.com/zh-CN/xpath/topic/6171/>

3:

XPath 1.0。

XMLXPath - XPathXPath。

Examples

```
<root xmlns="http://test/">
  <element xmlns:example="http://foobar/">
    <example:hello_world attribute="another example" />
  </element>
</root>
```

/rootroot。 <root xmlns="http://test/">。

```
/*[namespace-uri() = 'http://test/' and local-name() = 'root']
```

<https://riptutorial.com/zh-CN/xpath/topic/2324/>

4:

1. • / path/ element [@attribute_name]
2. • // * [@]
3. • / path to / element [@ attribute_name ='search value']
• / path to / element [@ attribute_name ="search value"]
4. • // * [@ attribute_name ="]
• // * [@ attribute_name =""]

@attribute_name

[@attribute_name]◦

starts-withlowercase◦

Examples

XML

```
<Galaxy>
  <name>Milky Way</name>
  <CelestialObject name="Earth" type="planet"/>
  <CelestialObject name="Sun" type="star"/>
</Galaxy>
```

XPATH

```
/Galaxy/*[@name]
```

```
//*[@name]
```

OUTPUT

```
<CelestialObject name="Earth" type="planet" />
<CelestialObject name="Sun" type="star" />
```

XML

```
<Galaxy>
  <name>Milky Way</name>
  <CelestialObject name="Earth" type="planet"/>
  <CelestialObject name="Sun" type="star"/>
</Galaxy>
```

XPATH

```
/Galaxy/*[@name='Sun']
```

```
//*[ @name='Sun']
```

OUTPUT

```
<CelestialObject name="Sun" type="star" />
```

XML

```
<Galaxy>
  <name>Milky Way</name>
  <CelestialObject name="Earth" type="planet"/>
  <CelestialObject name="Sun" type="star"/>
</Galaxy>
```

XPATH

```
/Galaxy/*[contains(@name, 'Ear')]
```

```
//*[contains(@name, 'Ear')]
```

```
/Galaxy/*[contains(@name, "Ear")]
```

OUTPUT

```
<CelestialObject name="Earth" type="planet" />
```

XML

```
<Galaxy>
  <name>Milky Way</name>
  <CelestialObject name="Earth" type="planet"/>
  <CelestialObject name="Sun" type="star"/>
</Galaxy>
```

XPATH

```
/Galaxy/*[contains(lower-case(@name), 'ear')]
```

```
//*[contains(lower-case(@name), 'ear')]
```

```
//*[contains(lower-case(@name), "ear")]
```

OUTPUT

```
<CelestialObject name="Earth" type="planet" />
```

XML

```
<Galaxy>
  <name>Milky Way</name>
  <CelestialObject name="Earth" type="planet"/>
  <CelestialObject name="Sun" type="star"/>
</Galaxy>
```

XPATH

```
/Galaxy/*[starts-with(lower-case(@name), 'ear')]
```

```
//*[starts-with(lower-case(@name), 'ear')]
```

OUTPUT

```
<CelestialObject name="Earth" type="planet" />
```

XML

```
<Galaxy>
  <name>Milky Way</name>
  <CelestialObject name="Earth" type="planet"/>
  <CelestialObject name="Sun" type="star"/>
</Galaxy>
```

XPATH

```
/Galaxy/*[ends-with(lower-case(@type), 'tar')]
```

```
//*[ends-with(lower-case(@type), 'tar')]
```

OUTPUT

```
<CelestialObject name="Sun" type="star" />
```

<https://riptutorial.com/zh-CN/xpath/topic/3096/>

5:

Examples

XML

```
<root>
  <element foobar="hello_world" />
  <element example="this is one!" />
</root>
```

```
/root/element[@foobar]
```

```
<element foobar="hello_world" />◦
```

XML

```
<root>
  <element foobar="hello_world" />
  <element example="this is one!" />
</root>
```

XPath

```
/root/element[@foobar = 'hello_world']
```

```
<element foobar="hello_world" />◦
```

```
/root/element[@foobar="hello_world"]
```

<https://riptutorial.com/zh-CN/xpath/topic/6488/>

6:

Examples

XML

```
<root>
  <element>hello</element>
  <another>
    hello
  </another>
  <example>Hello, <nested> I am an example </nested>.</example>
</root>
```

XPath

```
//*[text() = 'hello']
```

```
<element>hello</element><another>◦ <another>hello◦
```

```
<element><another>
```

```
//*[normalize-space(text()) = 'hello']
```

```
//*[normalize-space() = 'hello']
```

◦ text()normalize-space◦

contains◦ <example>

```
//example[contains(text(), 'Hello')]
```

/.text()◦ .◦

```
//example[. = 'Hello, I am an example .']
```

```
//example//text()
```

- “ ”
- “”
- “”

string

```
string(//example[1])
```



```
string(//example)
```

◦

stringXPath 1.0◦

```
string(/root/*)
```

<https://riptutorial.com/zh-CN/xpath/topic/1903/>

7:

Examples

XML

```
<Students>
  <Student>
    <Name>
      <First>Ashley</First>
      <Last>Smith</Last>
    </Name>
    <Grades>
      <Exam1>A</Exam1>
      <Exam2>B</Exam2>
      <Final>A</Final>
    </Grades>
  </Student>
  <Student>
    <Name>
      <First>Bill</First>
      <Last>Edwards</Last>
    </Name>
    <Grades>
      <Exam1>A</Exam1>
    </Grades>
  </Student>
</Students>
```

XPath

2

```
//Student[count(./Grades/*) > 1]
```

```
<Student>
  <Name>
    <First>Ashley</First>
    <Last>Smith</Last>
  </Name>
  <Grades>
    <Exam1>A</Exam1>
    <Exam2>B</Exam2>
    <Final>A</Final>
  </Grades>
</Student>
```

XML

```
<Students>
  <Student>
    <Name>
      <First>Ashley</First>
```

```
        <Last>Smith</Last>
    </Name>
    <Grades>
        <Exam1>A</Exam1>
        <Exam2>B</Exam2>
        <Final>A</Final>
    </Grades>
</Student>
<Student>
    <Name>
        <First>Bill</First>
        <Last>Edwards</Last>
    </Name>
    <Grades>
        <Exam1>A</Exam1>
    </Grades>
</Student>
</Students>
```

XPath

Exam2

```
//Student [./Grades/Exam2]
```

```
//Student [./Exam2]
```

```
<Student>
  <Name>
    <First>Ashley</First>
    <Last>Smith</Last>
  </Name>
  <Grades>
    <Exam1>A</Exam1>
    <Exam2>B</Exam2>
    <Final>A</Final>
  </Grades>
</Student>
```

<https://riptutorial.com/zh-CN/xpath/topic/6504/>

8:

- path_to_node

- 1.
2. NaN0

Examples

XML

```
<Animal>
  <legs>4</legs>
  <eyes>2</eyes>
  <horns>2</horns>
  <tail>1</tail>
</Animal>
```

XPATH

```
boolean(/Animal/tusks)
```

OUTPUT

```
false
```

XPATH

```
<Animal>
  <legs>4</legs>
  <eyes>2</eyes>
  <horns>2</horns>
  <tail>1</tail>
</Animal>
```

XPATH

```
boolean(/Animal/horns)
```

OUTPUT

```
true
```

XML

```
<House>
  <LivingRoom>
    <plant name="rose"/>
  </LivingRoom>
</House>
```

```
</LivingRoom>  
<TerraceGarden>  
  <plant name="passion fruit"/>  
  <plant name="lily"/>  
  <plant name="golden duranta"/>  
</TerraceGarden>  
</House>
```

XPATH

```
boolean (/House//plant)
```

OUTPUT

```
true
```

<https://riptutorial.com/zh-CN/xpath/topic/7432/>

9:

- path_to_node /
- stringpath_to_node="

- 1.
2. NaN0

String。

Examples

Deborah

XML

```
<Deborah>
  <address>Dark world</address>
  <master>Babadi</master>
  <ID>#0</ID>
  <colour>red</colour>
  <side>evil</side>
</Deborah>
```

XPATH

```
boolean (/Deborah/master/text ())
```

```
string (/Deborah/master) != ''
```

OUTPUT

```
true
```

Dobby

XML

```
<Dobby>
  <address>Hogwartz</address>
  <master></master>
  <colour>wheatish</colour>
  <side>all good</side>
</Dobby>
```

XPATH

```
boolean(/Dobby/master/text())
```

```
string(/Dobby/master) != ''
```

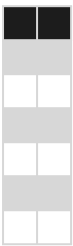
OUTPUT

```
false
```

<https://riptutorial.com/zh-CN/xpath/topic/7445/>

10:

1. • / node / ancestor :: node
2. • / node / ancestor :: ancestor_name
3. • / node / parent :: node
4. • / node to following-sibling :: node
5. • / node / following-sibling :: sibling_name
6. • / node to previous-sibling :: node
7. • / node / preceding-sibling :: sibling_name
8. • / node / child :: node
9. • / node / child :: chid_name
10. • / node / descendant :: node
11. • / path to node / descendant :: descendant_name



◦

Examples

XML

```
<GrandFather name="Bardock" gender="male" spouse="Gine">
  <Dad name="Goku" gender="male" spouse="Chi Chi">
    <Me name="Gohan" gender="male" />
    <brother name="Goten" gender="male" />
  </Dad>
</GrandFather>
```

XPATH

```
//Me/ancestor::node()
```

OUTPUT

```
<GrandFather name="Bardock" gender="male" spouse="Gine">
  <Dad name="Goku" gender="male" spouse="Chi Chi">
    <Me name="Gohan" gender="male" />
    <brother name="Goten" gender="male" />
  </Dad>
</GrandFather>
<Dad name="Goku" gender="male" spouse="Chi Chi">
  <Me name="Gohan" gender="male" />
  <brother name="Goten" gender="male" />
```



```
</Dad>
```

XML

```
<GrandFather name="Bardock" gender="male" spouse="Gine">  
  <Dad name="Goku" gender="male" spouse="Chi Chi">  
    <Me name="Gohan" gender="male" />  
    <brother name="Goten" gender="male" />  
  </Dad>  
</GrandFather>
```

XPATH

```
//Me/ancestor::Dad
```

```
//Me/parent::node()
```

OUTPUT

```
<Dad name="Goku" gender="male" spouse="Chi Chi">  
  <Me name="Gohan" gender="male" />  
  <brother name="Goten" gender="male" />  
</Dad>
```

XML

```
<GrandFather name="Bardock" gender="male" spouse="Gine">  
  <Dad name="Goku" gender="male" spouse="Chi Chi">  
    <Me name="Gohan" gender="male" />  
    <brother name="Goten" gender="male" />  
  </Dad>  
</GrandFather>
```

XPATH

```
//Me/ancestor::GrandFather
```

```
//Me/parent::node()/parent::node()
```

OUTPUT

```
<GrandFather name="Bardock" gender="male" spouse="Gine">  
  <Dad name="Goku" gender="male" spouse="Chi Chi">  
    <Me name="Gohan" gender="male" />  
    <brother name="Goten" gender="male" />  
  </Dad>  
</GrandFather>
```

XML

```
<GrandFather name="Bardock" gender="male" spouse="Gine">
  <Dad name="Goku" gender="male" spouse="Chi Chi">
    <brother name="Goten" gender="male" />
    <Me name="Gohan" gender="male" />
    <brother name="Goten" gender="male" />
  </Dad>
</GrandFather>
```

XPATH

```
//Me/following-sibling::brother
```

OUTPUT

```
<brother name="Goten" gender="male" />
```

Parashurama

XML

```
<Dashavatar>
  <Avatar name="Matsya"/>
  <Avatar name="Kurma"/>
  <Avatar name="Varaha"/>
  <Avatar name="Narasimha"/>
  <Avatar name="Vamana"/>
  <Avatar name="Balabhadra"/>
  <Avatar name="Parashurama"/>
  <Avatar name="Rama"/>
  <Avatar name="Krishna"/>
  <Avatar name="Kalki"/>
</Dashavatar>
```

XPATH

```
//Avatar[@name='Parashurama']/preceding-sibling::node()
```

OUTPUT

```
<Avatar name="Matsya"/>
<Avatar name="Kurma"/>
<Avatar name="Varaha"/>
<Avatar name="Narasimha"/>
<Avatar name="Vamana"/>
<Avatar name="Balabhadra"/>
```

Parashurama

XML

```
<Dashavatar>
```

```
<Avatar name="Matsya"/>
<Avatar name="Kurma"/>
<Avatar name="Varaha"/>
<Avatar name="Narasimha"/>
<Avatar name="Vamana"/>
<Avatar name="Balabhadra"/>
<Avatar name="Parashurama"/>
<Avatar name="Rama"/>
<Avatar name="Krishna"/>
<Avatar name="Kalki"/>
</Dashavatar>
```

XPATH

```
//Avatar[@name='Parashurama']/following-sibling::node()
```

OUTPUT

```
<Avatar name="Rama" />
<Avatar name="Krishna" />
<Avatar name="Kalki" />
```

Parusharama

XML

```
<Dashavatar>
  <Avatar name="Matsya"/>
  <Avatar name="Kurma"/>
  <Avatar name="Varaha"/>
  <Avatar name="Narasimha"/>
  <Avatar name="Vamana"/>
  <Avatar name="Balabhadra"/>
  <Avatar name="Parashurama"/>
  <Avatar name="Rama"/>
  <Avatar name="Krishna"/>
  <Avatar name="Kalki"/>
</Dashavatar>
```

XPATH

```
//Avatar[@name='Parashurama']/following-sibling::Avatar |
//Avatar[@name='Parashurama']/preceding-sibling::Avatar
```

OUTPUT

```
<Avatar name="Matsya" />
<Avatar name="Kurma" />
<Avatar name="Varaha" />
<Avatar name="Narasimha" />
<Avatar name="Vamana" />
<Avatar name="Balabhadra" />
<Avatar name="Rama" />
<Avatar name="Krishna" />
```

```
<Avatar name="Kalki" />
```

House

XML

```
<House>  
  <Rooms>10</Rooms>  
  <People>4</People>  
  <TVs>4</TVs>  
  <Floors>2</Floors>  
</House>
```

XPATH

```
/House/child::node()
```

OUTPUT

```
<Rooms>10</Rooms>  
<People>4</People>  
<TVs>4</TVs>  
<Floors>2</Floors>
```

Room

XML

```
<House>  
  <numRooms>4</numRooms>  
  <Room name="living" />  
  <Room name="master bedroom" />  
  <Room name="kids' bedroom" />  
  <Room name="kitchen" />  
</House>
```

XPATH

```
/House/child::Room
```

```
/House/*[local-name()='Room']
```

OUTPUT

```
<Room name="living" />  
<Room name="master bedroom" />  
<Room name="kids' bedroom" />  
<Room name="kitchen" />
```

XML

```
<House>
  <numRooms>4</numRooms>
  <Floor number="1">
    <Room name="living"/>
    <Room name="kitchen"/>
  </Floor>
  <Floor number="2">
    <Room name="master bedroom"/>
    <Room name="kids' bedroom"/>
  </Floor>
</House>
```

XPATH

```
/House/descendant::Room
```

OUTPUT

```
<Room name="living" />
<Room name="kitchen" />
<Room name="master bedroom" />
<Room name="kids' bedroom" />
```

<https://riptutorial.com/zh-CN/xpath/topic/6495/>

11:

•



◦

Examples

XML

```
<Goku>
  <child name="Gohan"/>
  <child name="Goten"/>
</Goku>
```

XPATH

```
count (/Goku/child)
```

OUTPUT

```
2.0
```

XML

```
<House>
  <LivingRoom>
    <plant name="rose"/>
  </LivingRoom>
  <TerraceGarden>
    <plant name="passion fruit"/>
    <plant name="lily"/>
    <plant name="golden duranta"/>
  </TerraceGarden>
</House>
```

XPATH

```
count (/House//plant)
```

OUTPUT

```
4.0
```

<https://riptutorial.com/zh-CN/xpath/topic/4463/>

12:

1. {path-to-parent} / name='search string']
2. // * [name="]



local-nameXPATH

Examples

LightDeviceSensor

XML

```
<Galaxy>
  <Light>sun</Light>
  <Device>satellite</Device>
  <Sensor>human</Sensor>
  <Name>Milky Way</Name>
</Galaxy>
```

XPATH

```
/Galaxy/*[local-name()='Light' or local-name()='Device' or local-name()='Sensor']
```

```
//*[local-name()='Light' or local-name()='Device' or local-name()='Sensor']
```

OUTPUT

```
<Light>sun</Light>
<Device>satellite</Device>
<Sensor>human</Sensor>
```

Light

XML

```
<Data>
  <BioLight>
    <name>Firefly</name>
    <model>Insect</model>
  </BioLight>
  <ArtificialLight>
    <name>Fire</name>
    <model>Natural element</model>
    <source>flint</source>
```

```
</ArtificialLight>
<SolarLight>
  <name>Sun</name>
  <model>Star</model>
  <source>helium</source>
</SolarLight>
</Data>
```

XPATH

```
/Data/*[contains(local-name(),"Light")]
```

```
//*[contains(local-name(),"Light")]
```

OUTPUT

```
<BioLight>
  <name>Firefly</name>
  <model>Insect</model>
</BioLight>
<ArtificialLight>
  <name>Fire</name>
  <model>Natural element</model>
  <source>flint</source>
</ArtificialLight>
<SolarLight>
  <name>Sun</name>
  <model>Star</model>
  <source>helium</source>
</SolarLight>
```

Star

XML

```
<College>
  <FootBall>
    <Members>20</Members>
    <Coach>Archie Theron</Coach>
    <Name>Wild cats</Name>
    <StarFootballer>David Perry</StarFootballer>
  </FootBall>
  <Academics>
    <Members>100</Members>
    <Teacher>Tim Jose</Teacher>
    <Class>VII</Class>
    <StarPerformer>Lindsay Rowen</StarPerformer>
  </Academics>
</College>
```

XPATH

```
/College/*/*[starts-with(local-name(),"Star")]
```



```
//*[starts-with(local-name(),"Star")]
```

OUTPUT

```
<StarFootballer>David Perry</StarFootballer>  
<StarPerformer>Lindsay Rowen</StarPerformer>
```

Ball

XML

```
<College>  
  <Football>  
    <Members>20</Members>  
    <Coach>Archie Theron</Coach>  
    <Name>Wild cats</Name>  
    <StarPlayer>David Perry</StarPlayer>  
  </Football>  
  <VolleyBall>  
    <Members>24</Members>  
    <Coach>Tim Jose</Coach>  
    <Name>Avengers</Name>  
    <StarPlayer>Lindsay Rowen</StarPlayer>  
  </VolleyBall>  
  <FoosBall>  
    <Members>22</Members>  
    <Coach>Rahul Mehra</Coach>  
    <Name>Playerz</Name>  
    <StarPlayer>Amanda Ren</StarPlayer>  
  </FoosBall>  
</College>
```

XPATH

```
/College/*[ends-with(local-name(),"Ball")]
```

```
//*[ends-with(local-name(),"Ball")]
```

OUTPUT

```
<Football>  
  <Members>20</Members>  
  <Coach>Archie Theron</Coach>  
  <Name>Wild cats</Name>  
  <StarPlayer>David Perry</StarPlayer>  
</Football>  
<VolleyBall>  
  <Members>24</Members>  
  <Coach>Tim Jose</Coach>  
  <Name>Avengers</Name>  
  <StarPlayer>Lindsay Rowen</StarPlayer>  
</VolleyBall>  
<FoosBall>  
  <Members>22</Members>
```

```
<Coach>Rahul Mehra</Coach>
<Name>Playerz</Name>
<StarPlayer>Amanda Ren</StarPlayer>
</FoosBall>
```

light

XML

```
<Galaxy>
  <Light>sun</Light>
  <Device>satellite</Device>
  <Sensor>human</Sensor>
  <Name>Milky Way</Name>
</Galaxy>
```

XPATH

```
/Galaxy/*[lower-case(local-name())="light"]
```

```
//*[lower-case(local-name())="light"]
```

OUTPUT

```
<Light>sun</Light>
```

<https://riptutorial.com/zh-CN/xpath/topic/3095/>

S. No	Contributors
1	xpath , Community , hielsnoppe , kjhughes , Vinay , Wolfgang Schindler
2	nwellnhof
3	4444 , Keith Hall
4	Keith Hall , miken32 , suchitra nair
5	Keith Hall
6	Keith Hall
7	Matthew
8	suchitra nair
9	suchitra nair
10	suchitra nair
11	suchitra nair
12	Dimitre Novatchev , suchitra nair