

Sikuli 가이드

이 가이드는 sikuli 공식 문서를 참고합니다.

이 안내서는 모든 실습 및 과제를 해결하기에 충분한 지식을 제공합니다.

또한, 기술 정보를 이해하기 쉬운 형태로 중요한 정보를 갖도록 고안되었습니다.

공식 문서 - <https://sikulix-2014.readthedocs.io/en/latest/#>

Sikuli 포럼(질문이나 문제가 있을 경우) - <https://answers.launchpad.net/sikuli>

1. 기초

`find(img_string OR Pattern)` - 주어진 이미지와 가장 잘 일치하는 주어진 영역 (또는 전체 화면)을 검색합니다. (문자열 또는 패턴 (문자열)). 일치하는 것이 없으면 `FindFailed` 예외(exception)가 발생합니다.

`exists(img_string OR Pattern)` - `find()`와 유사한 인수를 사용하여 일치 항목을 검색합니다. 일치 항목이 발견된 경우 - 일치 항목을 반환하고 그렇지 않으면 `None`을 반환합니다.

`click(img_string OR Pattern OR Match OR Location)` - 이미지의 일치 항목을 찾습니다. (문자열 또는 패턴이 인수로 사용된 경우) 일치한 항목을 마우스 왼쪽 버튼으로 클릭합니다.

예) `click(image) = click(Pattern(image)) = click(find(Pattern(image))) = click(Match) = click(Location)`

`doubleClick(img_string OR Pattern OR Match)` - `click()`과 동일하지만 두 번 클릭 합니다.

`rightClick(img_string OR Pattern OR Match)` - `click()`과 동일하지만 마우스 왼쪽 클릭 대신 마우스 오른쪽 버튼 클릭을 합니다.

`dragDrop(img_string OR Pattern OR Match, img_string OR Pattern OR Match)` - 이미지의 일치 항목을 찾습니다. (필요한 경우) 첫 번째 일치 항목 위치에서 두 번째 일치 항목 위치로 끌어서 놓기(drag and Drop)를 시뮬레이션합니다.

2. 이미지 설정

Matching Preview - 현재 이미지에서 검색되는 모든 일치 항목을 표시합니다.

이 탭에서는 슬라이더를 사용하여 이미지의 유사성을 수정할 수 있습니다.

`Pattern(path_to_image).similar(int)`

Target Offset - 이미지의 현재 오프셋을 표시한다.

`Pattern(path_to_image).targetOffset(x, y)`

3. 영역(Region)

영역은 화면에서 직사각형 영역으로 x, y, w, h (x 및 y 는 영역의 왼쪽 위 모서리에 대한 좌표, w -너비 (픽셀), h -높이 (픽셀))로 정의됩니다. `click()`, `find()`, `dragDrop()` 등과 같은 기본 메소드는 테스트 범위를 좁히기 위해 영역에서 사용됩니다. (예 : `someReg.click()`) 영역 지정 메소드를 사용하지 않으면 기본 영역(전체화면)이 사용됩니다. 영역 자체는 화면(Screen) 클래스에서 가져오지만 다중 모니터 설정을 사용하지 않으면 사용할 필요가 없습니다. (모두 기본 화면에서 모든 것이 호출됨)

영역 구성 : `r = Region(x, y, w, h)`

`find()`에 의해 리턴 된 일치 항목(match)도 영역으로 사용할 수 있습니다. 일치 항목(match) (영역)의 좌표는 다음과 같이 추출 할 수 있습니다 : `m.x`, `m.y`.

`findAll(img_string OR Pattern)` - `find()`와 동일하지만 발견된 모든 일치 항목을 반복자(iterator) 객체로 반환합니다. 찾지 못한 경우 `FindFailed`가 발생합니다.

예시 :

```
matches = findAll(...) for m in matches:  
m.click()
```

`findAllList(img_string OR Pattern)` - `findAll()`과 동일하지만 반복자 객체 대신 목록을 반환합니다. 일치하는 항목이 없으면 `FindFailed`를 발생하지 않는 대신 빈 목록을 반환합니다.

`wait(img_string OR Pattern, seconds)` - 주어진 이미지가 영역에 나타날 때까지 주어진 시간 동안 기다립니다. (시간이 지정되지 않으면 영원히 기다립니다)

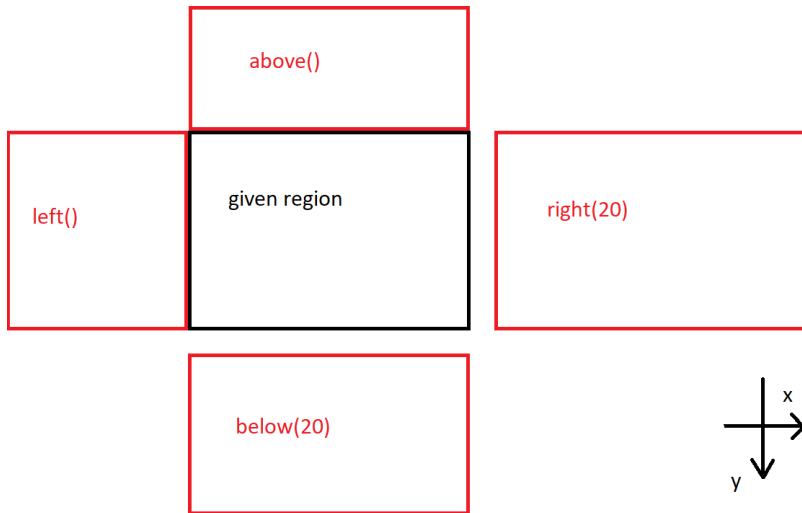
`waitVanish(img_string OR Pattern, seconds)` - `wait()`와 동일하지만 지정된 이미지가 영역에서 사라질 때까지 기다립니다.

`highlight(seconds)` - 지정된 시간 동안 영역을 강조 표시하여 빨간색 테두리를 표시합니다. 테스트 디버깅에 좋습니다.

`hover(img_string OR Pattern)` - `dragDrop()`과 동일하지만, 클릭을 수행하지 않습니다.

`above(range)`, `below(range)`, `left(range)`, `right(range)` - 주어진 영역과 관련하여 정의된 새 영역을 만듭니다. (아래 그림을 참조)

`above()`의 경우 : 지정된 영역(given region) 위쪽 테두리의 다음 행(row)입니다. (범위가 지정되지 않은 경우)



`grow(range)` - 4방향 모두에서 주어진 영역을 확장 (범위 > 0) 또는 축소 (범위 < 0) 합니다. `grow` (너비, 높이) 및 `grow` (왼쪽, 오른쪽, 위쪽, 아래쪽)으로도 사용할 수 있습니다.

4. 텍스트 조작

`collectWords()` - 영역에 포함된 텍스트를 왼쪽 상단에서 오른쪽 하단 순으로 일치하는 목록으로 추출합니다. 일치하는 텍스트를 얻으려면 `getText()`를 사용하십시오.

경고! 여기서 단어는 문자열을 의미하므로, 기호나 공백으로 구분된 것은 단어가 됩니다.

`collectLines()` - `collectWords()`와 동일하지만, 단어의 행을 찾는다.(finds lines of words)

`collectWordsText()` and `collectLinesText()` - 위 둘은 같은 작업을 수행하지만, 일치 항목 대신 문자열 목록을 반환합니다.

`type(string)` - 키보드에서 키 입력을 시뮬레이션한다.

`paste(string)` - 문자열을 한 번에 붙여넣습니다. 클립 보드를 비우는데 유용 할 수 있습니다.

그 외 다른 클래스와 함수들 :

`Pattern(img_string)` - 이미지에서 생성되어 `find()`, `click()` 등의 함수에 전달되는 객체

`Location(x, y)` - `click()`, `Drop()` 등을 특정 지점으로 드래그하는 데 사용할 수 있습니다.

`click("image.png")`을 사용하면 일치 항목의 x와 y에서 `Location()`을 만든 다음 클릭하는 데 사용됩니다.

`Env.getMouseLocation()` - 마우스의 현재 위치를 `Location()`으로 반환합니다.

`Env.getClipboard()` - 현재 복사된 문자열을 반환합니다.

5. 발생할 수 있는 문제

너무 작은 영역을 사용하면 이 오류가 발생합니다.

```
CvException: CvException [org.opencv.core.CvException: cv::Exception: OpenCV(3.4.2) C:\build\3_4_winpack-  
bindings-win64-vc14- static\opencv\modules\imgproc\src\templmatch.cpp:1107: error: (-215:Assertion failed)  
_img.size().height <= _templ.size().height && _img.size().width <= _templ.size().width in function  
'cv::matchTemplate']
```

이미지가 화면에 분명하게 표시되었음에도 FindFailed 오류가 발생했을 시 :
화면에서 변경한 후에는 일반적으로 `time.sleep ()` (0.2 초) 또는 `wait ()`를 사용하는 것이 좋습니다. 변경
사항은 Sikuli의 이미지 인식 알고리즘에 혼란을 주기 때문입니다.