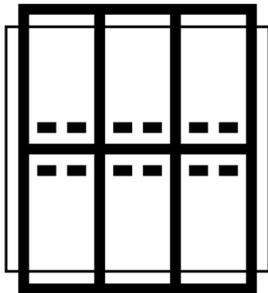


# PRKNG\_sqr / Users Manual (Kor)

For App Version: 1.0.0.2 / 20220823

App Developers: Woojae Sung @ SA:PN:DA ([www.sapnda.com/da](http://www.sapnda.com/da))

## PRKNG\_sqr [Parking Square]

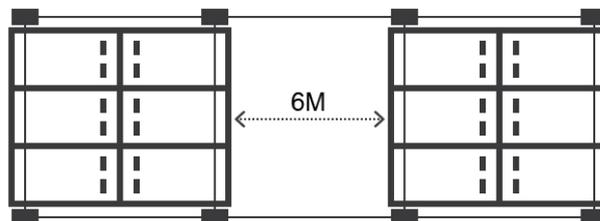


PRKNG 플러그인을 사용해 주셔서 감사합니다.

PRKNG\_sqr [Parking Square]는 정사각형 그리드를 활용하여 주어진 대지 경계내에 주차장 평면을 실시간으로 계산합니다.

정사각형 그리드는 어떠한 대지 형상에도 유연하게 적용될 수 있으며, 양방향의 구조설계 및 모듈러 시공에도 큰 장점이 있습니다.

이러한 정사각형 그리드는 이미 많은 나라에서 대규모의 지하 주차장의 설계과정에 활용되어 오고 있습니다. 하지만, 본 플러그인은 대한민국의 아파트 지하주차장 설계 방식에 기본을 두고 있습니다. 예를 들어 8.1미터의 정사각형 그리드는 600mm 두께의 기둥 사이에 2.5 x 5미터인 기본형 주차구획을 6개 배치 할 수 있습니다. 또한 주차장법에 명시되어 있듯이 주차 그리드 간 최소 6미터의 통로폭을 확보 할 수 있습니다.



사용법은 아주 쉽고 간단합니다. 사용자는 주차장의 경계를 설정하고, 그외 선택적인 몇가지의 정보만을 입력하면 됩니다. 또한 사용자는 디자인의 의도 및 필요성에 따라 정사각형

그리드의 크기, 주차구획의 크기 및 기둥의 크기를 임의로 수정할 수 있으며, 이는 실시간으로 주차평면에 반영됩니다.

## 설치

**PRKNG\_sqr** 는 Mac과 PC 기반의 Rhino3d (v7 이상)을 모두 지원합니다.

플러그인을 설치하는 방식은 아래와 같이 세가지가 있습니다.

- **Rhino3d의 package manager**

- Rhino3d를 열고 커맨드라인에 PackageManager를 입력합니다.
- 탐색영역에 parkng을 검색하고, **PRKNG\_sqr** 를 선택합니다.
- 그리고 설치 버튼을 누른후 화면의 지시에 따릅니다.
- Rhino3d와 Grasshopper를 종료후 다시 시작합니다.

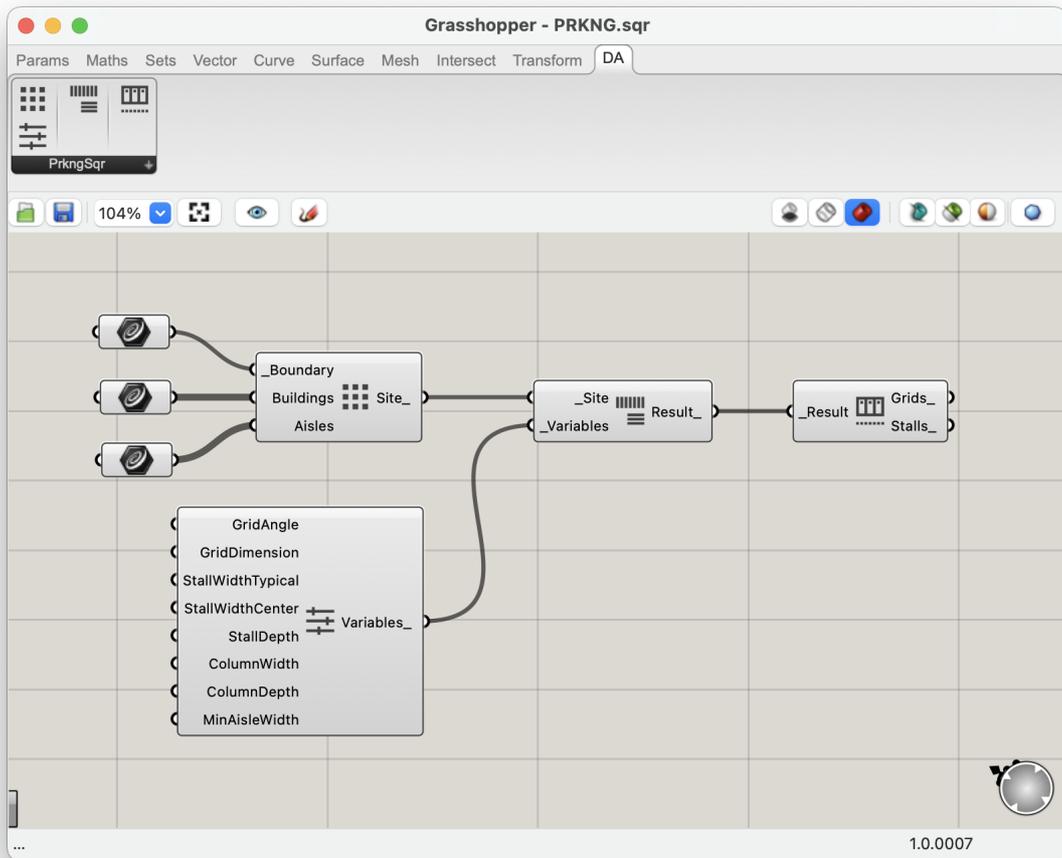
- **Food4Rhino**

- [Food4Rhino](#) 웹사이트를 방문합니다.
- Grasshopper Apps 메뉴를 선택합니다.
- parking square 를 검색합니다.
- 다운로드 후 화면의 지시를 따릅니다.
- 맥에서는 이 방식이 작동하지 않는 경우가 있습니다. 맥 사용자는 다른 방법으로 설치할 것을 권장합니다.

- **Manual installation**

- PackageManager 를 통한 방법이 권장되나, 이전 방식인 수동설치도 가능합니다.
- Rhino3d와 Grasshopper를 엽니다.
- Grasshopper 메뉴에서 File > Special Folders > Components Folder를 선택합니다.
- 다운로드 받은 gha파일을 해당 폴더에 복사합니다.
- 파일의 속성 정보에서 unblock 상태인지 확인합니다.
- Rhino3d와 Grasshopper를 종료후 다시 시작합니다.

## 개요



성공적으로 설치가 되었다면, **DA**[Design-Autonomy] 헬프에 **PRKNG\_sqr** 가 나타납니다. 기본 버전은 네개의 컴포넌트를 가지고 있습니다: Get Site, Get Variables, Compute Parking, Visualize Parking. 네 단계의 간단한 과정을 거쳐 사용자는 그리드와 주차구획의 리스트 결과를 아주 짧은 시간에 얻을 수 있습니다.

## 컴포넌트

### 규칙

보다 직관적인 사용방식을 위해, LB/HB에서 사용하는 방식을 차용하여 변수의 이름을 구성 하였습니다. 컴포넌트의 변수 명이 언더스코어로 시작을 한다면, 이는 해당 변수가 컴포넌트의 필수 입력값임을 의미합니다. 언더스코어가 없다면, 이는 기본 값이 이미 정의되어 있으나, 사용자의 필요에 따라 변경할 수 있는 선택적인 변수임을 의미합니다. 컴포넌트들을 연결하는 방식은, 동일 변수명(언더스코어를 제외한)들을 차례로 연결하면 됩니다.

### 단위

플러그인의 단위는 미터로 설정되어 있습니다. 미터 이외의 단위를 사용한다면, 아래 설명할 “Get Variables” 컴포넌트에 사용자가 지정한 단위로 환산한 값을 입력하면 됩니다.



## Get Site Information

이 컴포넌트는 대지관련 정보를 입력받고, 이를 다음단계로 전달하는 역할을 합니다.

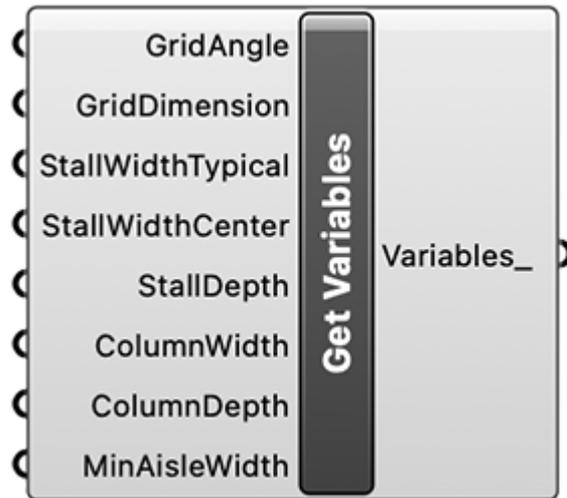


- **\_Boundary**: 반드시 닫혀있는 폴리라인을 입력해야 하며, 이는 주차장의 경계를 의미합니다. 하지만 계산 과정에 따라 입력한 폴리라인과 다소 상이한 최종 경계를 얻을 수 있습니다.
- **Buildings**: 선택적입력값. 주차장 경계내에 존재하는 건물의 경계를 의미하며, 닫혀있는 하나 이상의 폴리라인을 입력받습니다.
- **Aisles**: 선택적입력값. 주차장 경계내를 지나는 동선을 표현하며, 열려있거나 닫혀있는 하나 이상의 폴리라인을 입력받습니다.
- **Site\_**: 계산의 결과값을 출력합니다.



## Get Variables

이 컴포넌트는 그리드, 주차구획, 그리고 기둥의 치수를 설정합니다. 모든 입력값은 선택적이며, 가장 효율적인 치수가 이미 입력되어 있습니다. 하지만 사용자는 필요에 따라 이 값들을 변경할 수 있습니다.

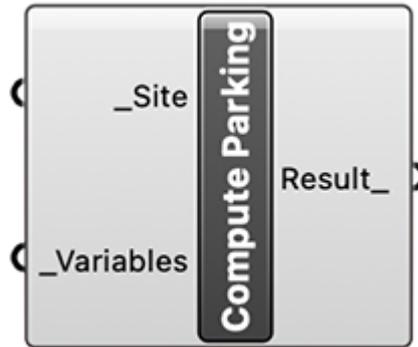


- **GridAngle:** 선택적입력값. 직교좌표를 벗어나는 이형의 대지를 사용할 때, 주축을 변경할 필요가 있을수 있습니다. 이때 주축의 방향과 그리드를 정렬하기 위해 이 값을 변경 할 수 있습니다. 각도는 degrees로 입력합니다.
- **GridDimnsion:** 선택적입력값. 기둥과 주차구획의 크기에 따라 그리드의 크기가 커지거나 작아져야 할 경우가 있습니다. 이 플러그인은 정사각형의 그리드를 사용하기때문에, 사용자는 단 하나만의 입력값을 통해 그리드의 크기를 변경할 수 있습니다.
- **StallWidthTypical / StallWidthCenter:** 선택적입력값. 주차구획은 기본형과 확장형으로 나뉘며, 기둥을 면해 기본형이 배치되고 중간에 확장형을 배치하게 됩니다. 기본 입력값은 각각 2.5미터와 2.6미터 입니다.
- **StallDepth:** 선택적입력값. 기본값은 5.0미터 입니다.
- **ColumnWidth:** 선택적입력값. 기본값은 0.6미터 입니다.
- **ColumnDepth:** 선택적입력값. 기본값은 0.6미터 입니다.
- **MinAisleWidth:** 선택적입력값. 대한민국 주차장 법에서는 자동차가 지나가는 통로가 적어도 6.0 미터 이상을 가지도록 규정하고 있습니다. 기본값은 6.0미터 입니다.
- **Variables\_:** 계산의 결과값을 출력합니다.



## Compute Parking Stalls

이 컴포넌트는 그리드 위에 주차 구획을 계산합니다.

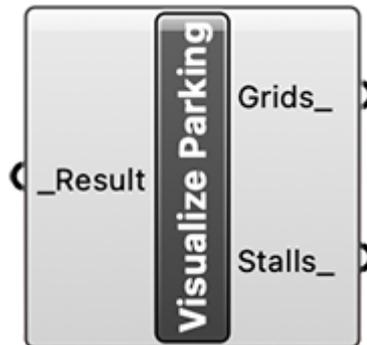


- **\_Site**: “Get Site” 의 결과값을 입력받습니다.
- **\_Variables**: “Get Variables” 의 결과값을 입력받습니다.
- **Result\_**: 계산의 결과값을 출력합니다.



## Visualize Parking Stalls

이 컴포넌트는 계산의 결과값을 시각화 하고, 이를 Rhino3d에서 활용가능한 형태로 출력합니다.



- **\_Result**: “Compute Parking” 의 결과값을 입력받습니다.
- **Grids\_**: 유효한 그리드 만을 출력합니다.
- **Stalls\_**: 유효한 주차구획을 출력합니다.

## Feedback / Bug Report

사용과정에 수정사항이나 버그가 발생한다면 [ws@sapnda.com](mailto:ws@sapnda.com) 으로 이메일을 보내주세요. 이메일을 작성시, 문제를 재현할 수 있는 모든 파일을 첨부하여야 합니다.