

## Inhoud

1. INLEIDING .....	5
1.1. Wat is Visual Programming? .....	5
Algoritmen in de Hand .....	5
Programmering gedefinieerd .....	6
2. HALLO DYNAMO!.....	9
2.2. De gebruikersinterface van Dynamo .....	9
Instellingen .....	13
Help .....	13
2.3. De werkruijnte.....	14
Tabbladen.....	15
Grafiek versus 3D Voorbeeld navigatie .....	15
Hallo muis!.....	16
In-Canvas zoeken.....	17
3. Dynamo voor Revit .....	18
3.1 De Revit-verbinding.....	18
Dynamo in Revit .....	19
3.2. Selecteren.....	20
3.3. Bewerken.....	23
Type en Instantieparameters .....	23
3.4. Maken.....	25
Adaptieve componenten.....	25
4. DE ANATOMIE VAN EEN VISUAL-PROGRAMMA .....	27
4.1. Knooppunten.....	27
Anatomie van een knooppunt.....	27
Poorten.....	27
Status.....	28
4.2. Draden.....	29
Programmeerstroom.....	29
Creëren van draden.....	29
Bewerken van draden .....	30
Draad Previews.....	31
4.3. Dynamo bibliotheek .....	32
Bibliotheek van bibliotheken.....	32
Veel gebruikte knooppunten.....	34
5.1. Data .....	36

Wat is data?.....	36
Pas op voor Null-waarden .....	37
Datastructuren .....	37
5.2. Wiskunde.....	38
Rekenkundige operatoren.....	38
Parametrische formule.....	39
Van formule naar geometrie .....	40
5.3. Logica.....	41
Booleans .....	42
Voorwaardelijke instructies.....	42
Een lijst filteren.....	43
5.4. De String .....	45
Tekenreeksen maken .....	45
Query Strings .....	45
Manipuleren van tekenreeksen .....	47
Werken met tekenreeksen.....	48
6. Ontwerpen met lijsten .....	52
6.1. Wat is een lijst? .....	52
Op nul gebaseerde Indices .....	52
Inputs en Outputs.....	53
Lacing.....	54

PREVIEW

# 1. INLEIDING

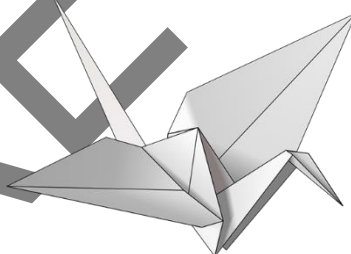
Vanaf zijn oorsprong als een add-on voor het gebouw informatie modelleren in Revit, is Dynamo om vele redenen volwassen geworden. Het is bovenal een platform, waarmee ontwerpers visueel programmeren kunnen verkennen, problemen oplossen en het maken hun eigen gereedschappen. Laten we beginnen aan onze reis met Dynamo met de context - wat is het en hoe kan ik het gebruiken?

## 1.1. Wat is Visual Programming?

Vaak gaat ontwerpen samen met het instellen van visuele, systemische of meetkundige relaties tussen de onderdelen van een ontwerp. Meestal worden deze relaties ontwikkeld door workflows die, door middel van regels, ons concept tot resultaat hebben. Misschien zonder het te weten, werken we met algoritmische definities door een stapsgewijze reeks acties van een fundamentele logica van input, verwerking en output. Met programmering kunnen we zo blijven werken, maar dan door onze algoritmen te formuleren.

### Algoritmen in de Hand

Terwijl het aanbieden van sommige krachtige mogelijkheden, de term **algoritme** enkele misvattingen mee dragen kan. Algoritmen kunnen leiden tot onverwachte, wild of leuke dingen, maar ze zijn niet magisch. In feite, zijn ze vrij duidelijk, in en van zichzelf. Laten we gebruik maken van een concreet voorbeeld zoals een origami-kraan. We beginnen met een vierkant stuk papier (input), volgen een reeks stappen (verwerking acties) vouwen, en resulteren in een kraanvogel (output).



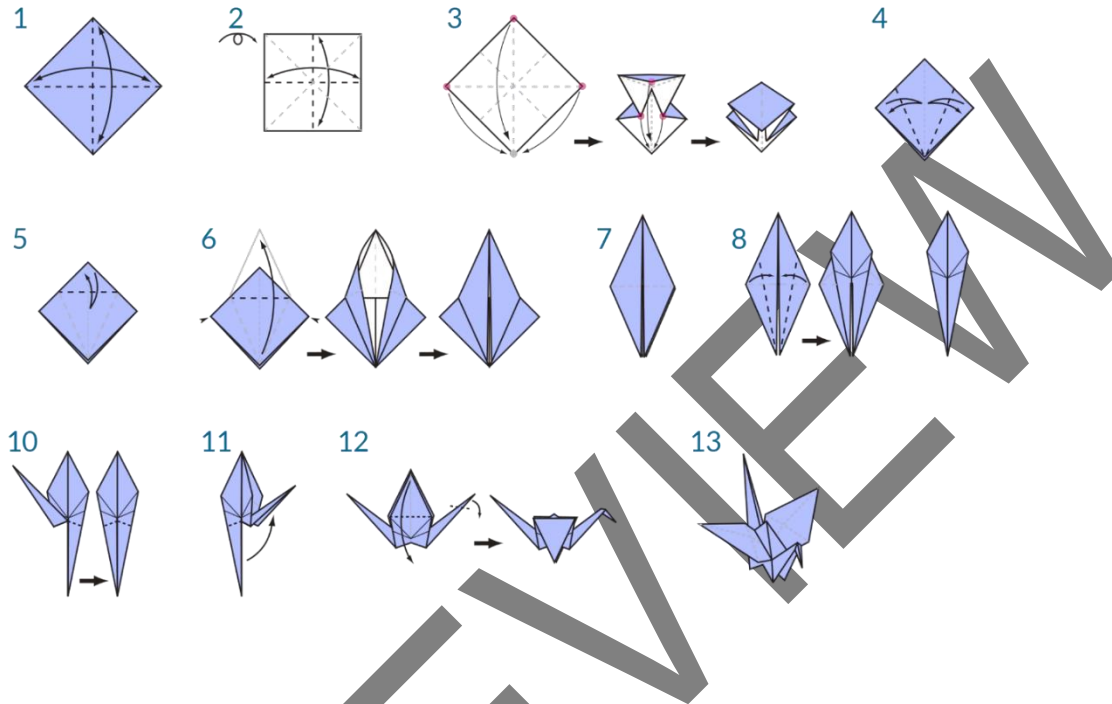
Dus waar is nu het algoritme? Het is de abstracte reeks stappen, die wij in een aantal manieren – hetzij tekstueel of grafisch kunnen vertegenwoordigen.

#### Tekstuele instructies:

1. Begin met een vierkant stuk papier, gekleurde kant naar boven. Vouw in het midden en open hem weer. Dan in de andere richting door het midden vouwen.
2. Draai het papier om naar de witte kant. Vouw het papier doormidden, vouw goed open en vouw vervolgens weer in de andere richting.
3. Met behulp van de plooien die u hebt aangebracht, kunt de hoeken van 3 punten van het model naar de 4e hoek vouwen. Duw het model plat.
4. Vouw de twee buitenste punten diagonaal naar binnen over het midden van de bovenste zijden en ontvouw weer.
5. Vouw de bovenkant van het model naar beneden, druk goed aan en ontvouw.
6. Open het onderste klepje van het model, vouw het naar boven en druk tegelijkertijd de zijanten van het model naar binnen. Duw het model goed plat.

7. Model draai en herhaal stap 4-6 aan de andere kant.
8. Vouw de buitenste flappen weer naar binnen met de vouwnaad op het midden van de bovenste zijden.
9. Herhaal aan de andere kant.
10. Beide "benen" van model opvouwen, kreuk zeer goed aanduwen, dan ontvouwen.
11. de "benen" langs de kreuken u zojuist hebt gemaakt via binnenzijde omhoog vouwen
12. Vouw één kant omlaag om een hoofd te maken, vouw daarna de vleugels naar beneden.
13. U hebt nu een kraanvogel.

### Grafische instructies:



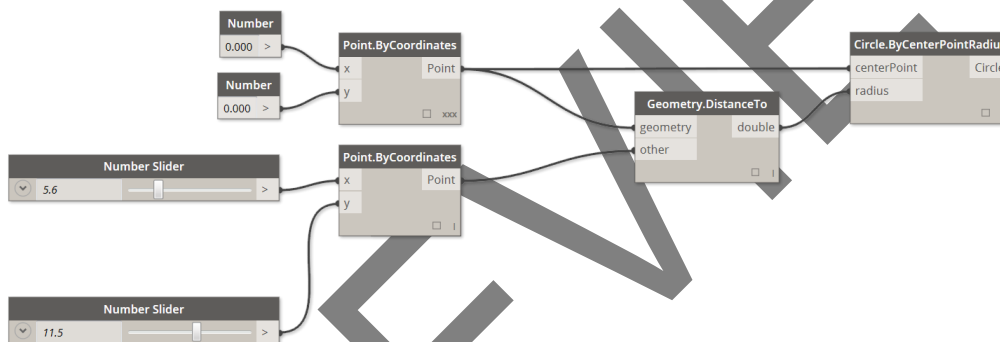
### Programmering gedefinieerd

Met behulp van een van deze sets van instructies zal het eindproduct resulteren in een kraanvogel, en als u deze stappen zelf heeft gevolgd, heeft u een algoritme toegepast. Het enige verschil met programmeren is de manier waarop we de set instructies formuleren. Programmering, verkorting van *Computer programming*, is de handeling van het formuleren van de verwerking van een reeks acties in een uitvoerbaar programma. Als we de bovenstaande instructies voor het maken van een kraanvogel omzetten naar een formaat dat onze computer kan lezen en uitvoeren, zijn wij aan het programmeren.

De sleutel tot en de eerste horde die we in de programmering vinden zullen, is dat we moeten vertrouwen op een soort abstractie om effectief te communiceren met onze computer. Die heeft de vorm van een willekeurig aantal programmering talen, zoals Javascript, Python en C. Als we een herhaalbare reeks instructies kunnen uitschrijven, zoals nodig voor de origami-kraanvogel, moeten wij die alleen vertalen voor de computer. We kunnen op deze manier met de computer een kraanvogel maken of zelfs verschillende kraanvogels die elk enigszins verschillen. Dit is de kracht van programmering - de computer elke taak of reeks taken, die wij hem aangeven, steeds opnieuw uitvoeren en onmiddellijk herhalen en zonder menselijke fouten.

Als je antwoord afbeeldingen bevatte, dan is **Visueel programmeren** zeker iets voor jou. Het proces is in wezen hetzelfde voor zowel programmeren als visuele programmeren. Ze maken gebruik van een en hetzelfde raamwerk van formuleren; echter definiëren we de instructies en de relaties van ons programma via een grafische (of "visuele") gebruikersinterface. In plaats van het typen van tekst gebonden syntaxis, verbinden we voorverpakte knooppunten met elkaar. Hier is een vergelijking van hetzelfde algoritme - "teken een cirkel door een punt" - geprogrammeerd met knooppunten versus code:

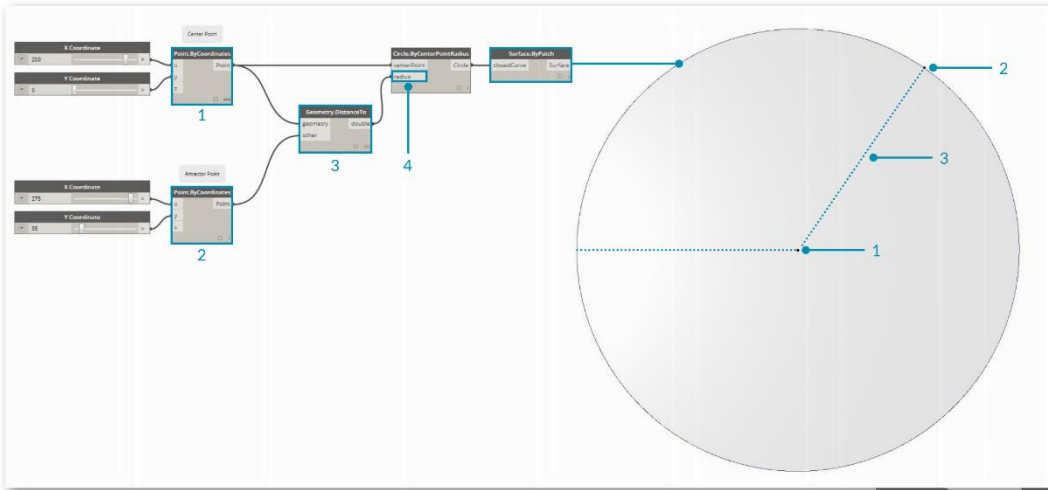
### Visual-programma:



### Tekstuele programma:

```
myPoint = Point.ByCoordinates(0.0,0.0,0.0);  
x = 5.6;  
y = 11.5;  
attractorPoint = Point.ByCoordinates(x,y,0.0);  
dist = myPoint.DistanceTo(attractorPoint);  
myCircle = Circle.ByCenterPointRadius(myPoint,dist);
```

De resultaten van ons algoritme:



De visuele karakteristiek voor het programmeren in een zodanige manier verlaagt de toetredingsdrempel en spreekt ontwerpers vaak aan. Dynamo valt in de Visual Programming groep, maar zoals we later zullen zien, kunnen we nog steeds tekstuele programmering in de toepassing gebruiken.

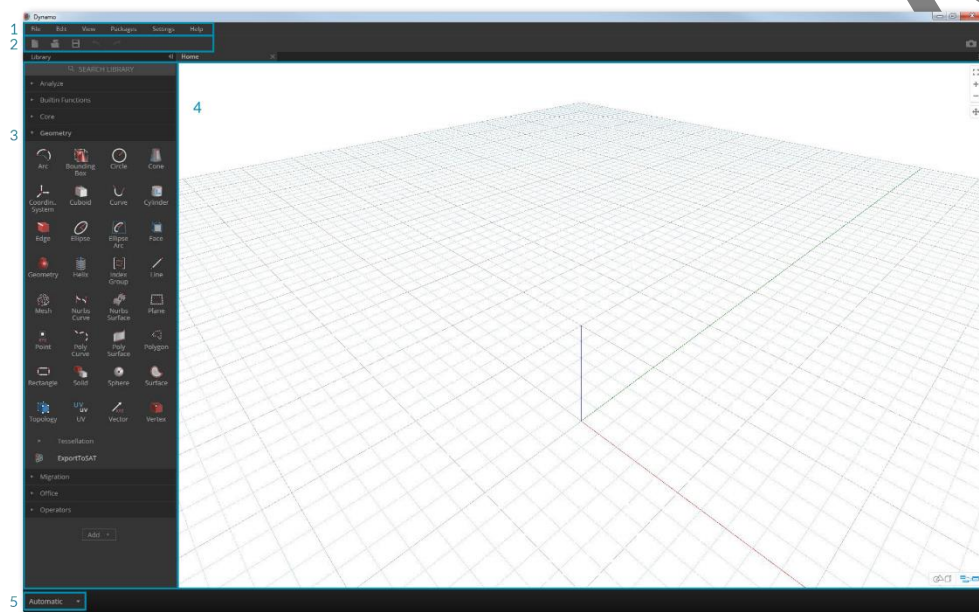
PREVIEW

## 2. HALLO DYNAMO!

In zijn kern is Dynamo een platform voor het visueel programmeren - het is een flexibele en uitbreidbare design tool. Omdat het als een stand-alone applicatie of als een add-on voor andere ontwerpsoftware kan werken, kunnen we een breed scala van creatieve workflows ontwikkelen. Laten we starten met het bekijken van de belangrijkste kenmerken van de interface.

### 2.1. De gebruikersinterface van Dynamo

De gebruikersinterface (UI) voor Dynamo is onderverdeeld in vijf grote delen, waarvan de grootste de werkruimte is waar we onze visuele programma's opstellen.

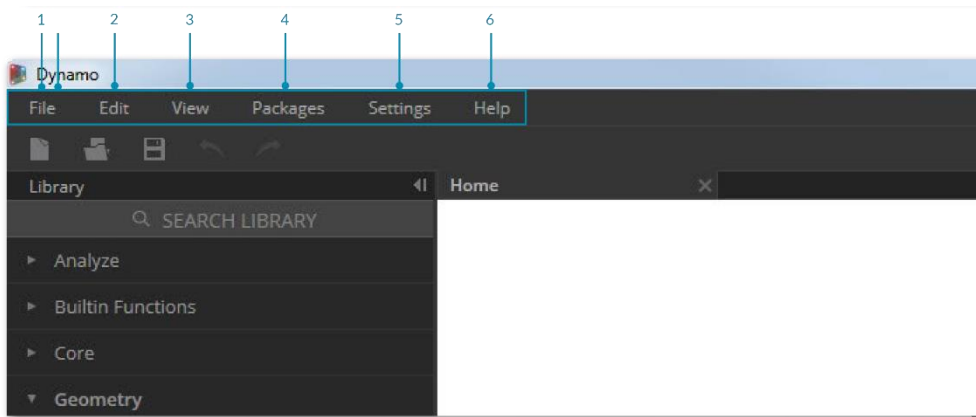


1. Menu's
2. Werkbalk
3. Bibliotheek
4. Werkruimte
5. Uitvoering Bar

Laten we dieper in de UI duiken en de functionaliteit van elke regio verkennen.

#### *Menu 's*

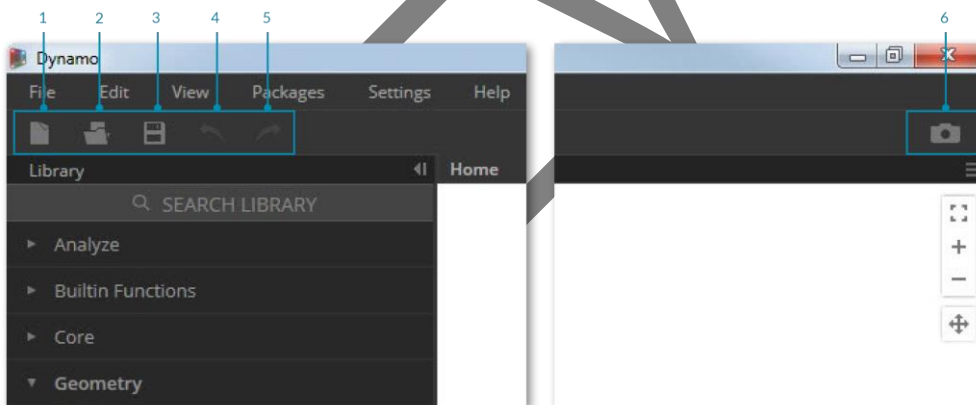
De Dropdown menu's zijn een geweldige plekken om de basisfunctionaliteit van de toepassing Dynamo te vinden. Zoals bij de meeste Windows-software, zijn acties met betrekking tot het beheren van bestanden en bewerkingen voor selectie en bewerken van de inhoud in de eerste twee menu's te vinden. De resterende menu's zijn meer specifiek voor Dynamo.



1. Bestand
2. Bewerken
3. Weergave
4. Pakketten
5. Instellingen
6. Help

### Werkbalk

Dynamo's werkbalk bevat een aantal knoppen voor snelle toegang tot het werken met bestanden zoals de Undo [Ctrl + Z] en Redo [Ctrl + Y] opdrachten. Aan de rechterkant is een andere knop die een momentopname van de werkruijme zal exporteren die uiterst nuttig is voor documentatie en delen.



1. Nieuw - maak een nieuw .dyn-bestand
2. Open - bestaande .dyn (werkruijme) of .dyf (aangepaste knooppunt) bestanden openen
3. Opslaan/Opslaan als - uw actieve .dyn of .dyf bestand opslaan
4. Ongedaan maken - uw laatste handeling ongedaan maken
5. Opnieuw - opnieuw de volgende actie uitvoeren
6. Exporteren als afbeelding - de zichtbare werkruijme als een PNG-bestand exporteren

### Bibliotheek

De bibliotheek bevat alle geladen knooppunten, inclusief de standaard geladen knooppunten die worden geleverd bij de installatie, evenals aangepaste knooppunten of pakketten. De knooppunten in de bibliotheek zijn hiërarchisch geordend in bibliotheken,

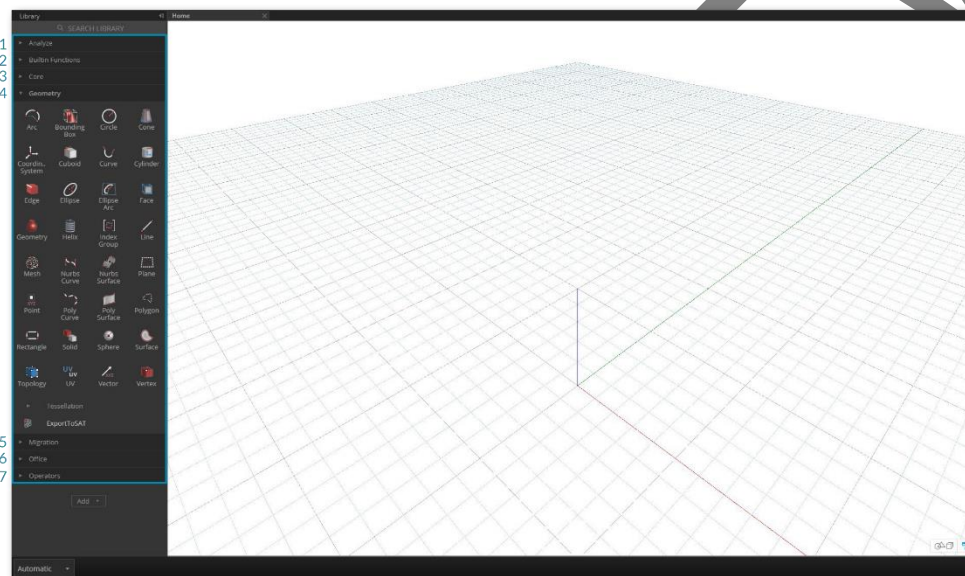


categorieën en, in voorkomende gevallen, sub categorieën op basis van of de knooppunten **gegevens maken**, een **actie** uitvoeren of **gegevens** verzamelen.

### Browsen

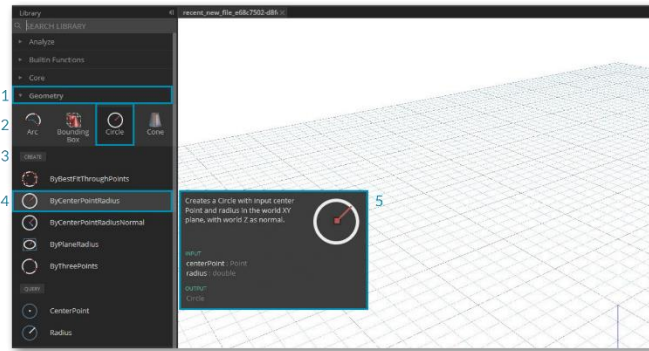
Standaard zal de **bibliotheek** acht soorten knooppunten bevatten. **Core** en **Geometry** zijn handige menu's om te mee beginnen en te verkennen omdat ze de grootste hoeveelheid knooppunten bevatten. Bladeren door deze categorieën is de snelste manier om de hiërarchie te begrijpen van wat we aan onze werkruimte kunt toevoegen en de beste manier om nieuwe knooppunten te leren kennen die u nog niet eerder hebt gebruikt.

We zullen ons nu richten op de standaard collectie van knooppunten, maar later kunnen we deze bibliotheek met aangepaste knooppunten, extra bibliotheken en de Package Manager uitbreiden.



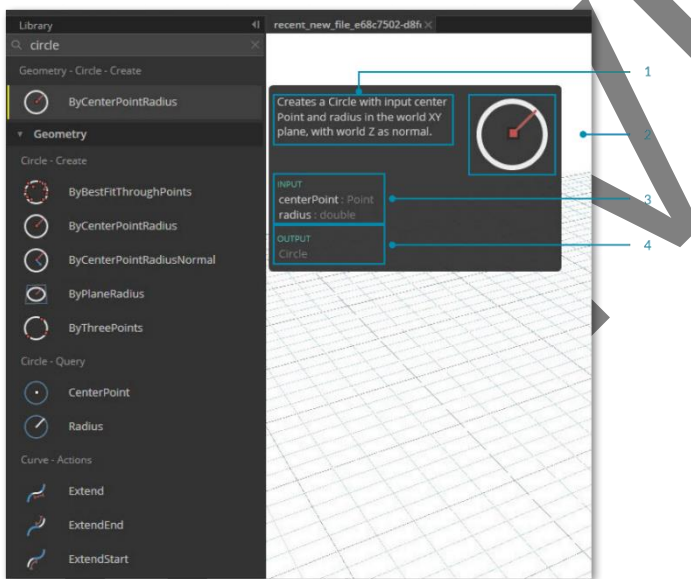
1. Analyseren
2. Ingebouwde functies
3. Kern
4. Meetkunde
5. Migratie
6. Office
7. Exploitanten

Bladeren door de bibliotheek kan door te klikken op de menu's. Klik op de Geometry > Circle. Let op het nieuwe gedeelte van het menu dat wordt geopend en specifiek **Create** en **Query**.



1. Bibliotheek
2. Categorie
3. Subcategorie: maken/acties/Query
4. Knooppunt

Als u vanuit hetzelfde menu als de cirkel uw muis over **ByCenterPointRadius** beweegt, verschijnt er een venster. Het venster onthult meer gedetailleerde informatie over het knooppunt naast de naam en het pictogram. Dit biedt ons een snelle manier om te begrijpen wat het knooppunt doet, wat het voor ingangen vereist en wat het zal geven als resultaat.



1. Icon - grotere versie van het pictogram in het bibliotheek-Menu
2. Beschrijving - de beschrijving in duidelijke taal van het knooppunt
3. Ingang(en) - naam, gegevenstype, en datastructuur
4. Uitgang(en) - gegevenstype en structuur

## Zoeken

Als u met relatieve zekerheid weet welk knooppunt u wilt toevoegen aan uw werkrumte, is **het zoekveld** uw beste vriend. Wanneer u geen instellingen bewerken of waarden opgeeft in de werkrumte, is de cursor altijd aanwezig in dit veld. Als u begint te typen, zal de Dynamo bibliotheek een selectie geven met beste opties (met tips voor waar en in welke categorie het knooppunt gevonden kan worden) en een lijst met alternatieven. Wanneer u