

## ENDPOINT MAPA\_ASTRAL

### Entrada de dados:

- Nome
- Data de nascimento
- Hora
- Minutos
- Local (cidade/estado) ou latitude/longitude

**OBS:** Estamos pensando em pedir a cidade para a pessoa, pegarmos a latitude/longitude e passarmos isso na chamada da API para evitar de ocorrer de pegar a cidade errada.

### Precisa retornar as seguintes informações, traduzidas para português:

- Planeta
- Qualidade
- Elemento
- Signo em que está
- Posição: valor está em decimal, converter para graus, segundos, minutos;
- Retrógrado
- Emoji

### Informação das casas, traduzidas para português:

- Casa
- Qualidade
- Elemento
- Signo em que está
- Posição: valor está em decimal, converter para graus, segundos, minutos;
- Emoji

### Exemplo dos dados:

```
NAME: Kanye
PLANET    POSITION

Sun:      Gem 17.599 in 12th House
Moon:     Pis 16.425 in 9th House
Mercury:  Tau 26.286 in 11th House
Venus:    Tau 2.032 in 10th House
Mars:     Tau 1.79 in 10th House
Jupiter:  Gem 14.607 in 11th House
Saturn:   Leo 12.799 in 2nd House
Uranus:   Sco 8.273 in 4th House
Neptune:  Sag 14.693 in 5th House
Pluto:    Lib 11.446 in 4th House
```

```
PLACIDUS HAUSES
House Cusp 1:  Can 17.996
House Cusp 2:  Leo 9.506
House Cusp 3:  Vir 4.022
House Cusp 4:  Lib 3.977
House Cusp 5:  Sco 9.393
House Cusp 6:  Sag 15.681
House Cusp 7:  Cap 17.996
House Cusp 8:  Aqu 9.506
House Cusp 9:  Pis 4.022
House Cusp 10: Ari 3.977
House Cusp 11: Tau 9.393
House Cusp 12: Gem 15.681
```

## Aspectos/trânsitos do mapa astral com o próprio mapa, traduzidos para o português:

- Entrada de dados:
  - Mapa astral natal

```

In [3]: from kerykeion import CompositeAspects, KrInstance

# Args: Name, year, month, day, hour, minuts, city, nation(optional)
first = KrInstance("Hildeberto", 1971, 9, 1, 9, 25, "Campina Grande", "Brazil")
second = KrInstance("Trânsitos", 1971, 9, 1, 9, 25, "Campina Grande", "Brazil")

name = CompositeAspects(first, second)
aspect_list = name.get_relevant_aspects()

# Filter out aspects where p1_name == p2_name
aspect_list = [aspect for aspect in aspect_list if aspect['p1_name'] != aspect['p2_name']]

data = {
    "aspects": []
}

# Use enumerate to add an index number to each aspect
for idx, aspect in enumerate(aspect_list):
    data["aspects"].append({
        "index": idx,
        "p1_name": aspect["p1_name"],
        "p1_abs_pos": aspect["p1_abs_pos"],
        "p2_name": aspect["p2_name"],
        "p2_abs_pos": aspect["p2_abs_pos"],
        "aspect": aspect["aspect"],
        "orbit": aspect["orbit"],
        "aspect_degrees": aspect["aspect_degrees"],
        "color": aspect["color"],
        "aid": aspect["aid"],
        "diff": aspect["diff"],
        "p1": aspect["p1"],
        "p2": aspect["p2"]
    })

print("p1_name\taspect\tp2_name")
for aspect in data["aspects"]:
    print(f"{aspect['p1_name']}\t{aspect['aspect']}\t{aspect['p2_name']}")
      
```



p1_name	aspect	p2_name
Sun	conjunction	Mercury
Sun	conjunction	Venus
Sun	square	Saturn
Mercury	conjunction	Sun
Mercury	square	Jupiter
Mercury	square	Neptune
Venus	conjunction	Sun
Venus	square	Saturn
Mars	quintile	Jupiter
Mars	trine	Saturn
Mars	trine	Uranus
Mars	quintile	Neptune
Jupiter	square	Mercury
Jupiter	quintile	Mars
Jupiter	opposition	Saturn
Jupiter	conjunction	Neptune
Jupiter	sextile	Pluto
Saturn	square	Sun
Saturn	square	Venus
Saturn	trine	Mars
Saturn	opposition	Jupiter
Saturn	trine	Uranus
Saturn	opposition	Neptune
Saturn	trine	Pluto
Uranus	trine	Mars
Uranus	trine	Saturn
Neptune	square	Mercury
Neptune	quintile	Mars
Neptune	conjunction	Jupiter
Neptune	opposition	Saturn
Neptune	sextile	Pluto
Pluto	sextile	Jupiter
Pluto	trine	Saturn
Pluto	sextile	Neptune

- Resultado esperado:
  - Descrição do planeta 1 (p1\_name)
  - Signo onde o Planeta 1 está
  - Casa onde o Planeta 1 está
  - Aspecto (quadratura, conjunção, etc...)
  - Descrição do planeta 2 (p2\_name)
  - Signo onde o Planeta 2 está
  - Casa onde o Planeta 2 está
  - Se possível incluir em qual casa/signo o planeta 1 e 2 estão. Ex: Sol em virgem na casa 1 conjunção Lua em virgem casa 1. Neste caso queremos que a informação possa ser puxada quebrada tipo assim:

	A	B	C	D	E	F	G
1	Planeta 1	Signo	Casa	Aspecto	Planeta 2	Signo	Casa
2	Sol	Virgem	Casa 1	conjunção	Lua	Virgem	Casa 1

- Importante: deve ser desconsiderado os aspectos iguais, por exemplo quando ocorrer SOL CONJUNÇÃO MARTE e MARTE CONJUNÇÃO SOL, deve apresentar somente 1.

## ENDPOINT ASPECTOS\_POR\_DATA

Serão utilizados os dados do mapa astral natal e o segundo objeto será fornecido os dados de local, hora, por padrão podemos deixar o de São Paulo (KrInstance("Trânsitos", 2023, 3, 15, 0, 00, "São Paulo", "Brazil")) e iremos fornecer um range de data de início e data fim. Onde buscaremos os aspectos relevantes por exemplo do SOL natal com JÚPITER, MARTE e SATURNO em trânsito no mapa da cidade de acordo com a data, desconsiderando os aspectos repetidos ex: SOL CONJUNÇÃO MARTE, MARTE

CONJUNÇÃO SOL, deve apresentar somente 1.

- **Entrada de dados:**

- Mapa astral natal
- Mapa do local trânsitos
- Range de data com Data Início e Data Fim
- Escolher qual(s) planeta(s) do mapa astral será considerado.
- Escolher qual(s) planeta(s) do mapa de trânsito será considerado.

- **Retorno esperado:**

- Descrição do Planeta 1
- Signo onde está o planeta 1
- Casa onde está o planeta 1
- Aspecto
- Descrição do Planeta 2
- Signo onde está o planeta 2
- Casa onde está o planeta 2
- Orbita (orbit)
- Graus (aspect\_degrees)

**Exemplo, porém está faltando os signos e casas:**

p1_name	aspect	p2_name	orbit	aspect_degrees
Sun	trine	Venus	3.639974148304617	120
Sun	trine	Mars	5.397734311087987	120
Sun	sextile	Saturn	3.9819372360229295	60
Moon	sextile	Jupiter	-3.357750739652431	60
Mercury	trine	Sun	1.0613186955964693	120
Mercury	trine	Moon	-7.284298402445131	120
Mercury	trine	Mars	-7.054479560317816	120

## ENDPOINT IMAGEM\_MAPA\_SVG

Será utilizado os dados do endpoint de MAPA\_ASTRAL para gerar a imagem;

```
: from kerykeion import KrInstance, MakeSvgInstance

# Args: Name, year, month, day, hour, minuts, city, nation(optional)
first = KrInstance("Luiza", 1993, 10, 30, 15, 15, "São Paulo", "Brazil")

name = MakeSvgInstance(first, chart_type="Natal")
name.makeSVG()
```

Luiza birth location: São Paulo, -23.5475, -46.63611

SVG Generated Correctly in: C:\Users\melaynnedias\LuizaNatalChart.svg

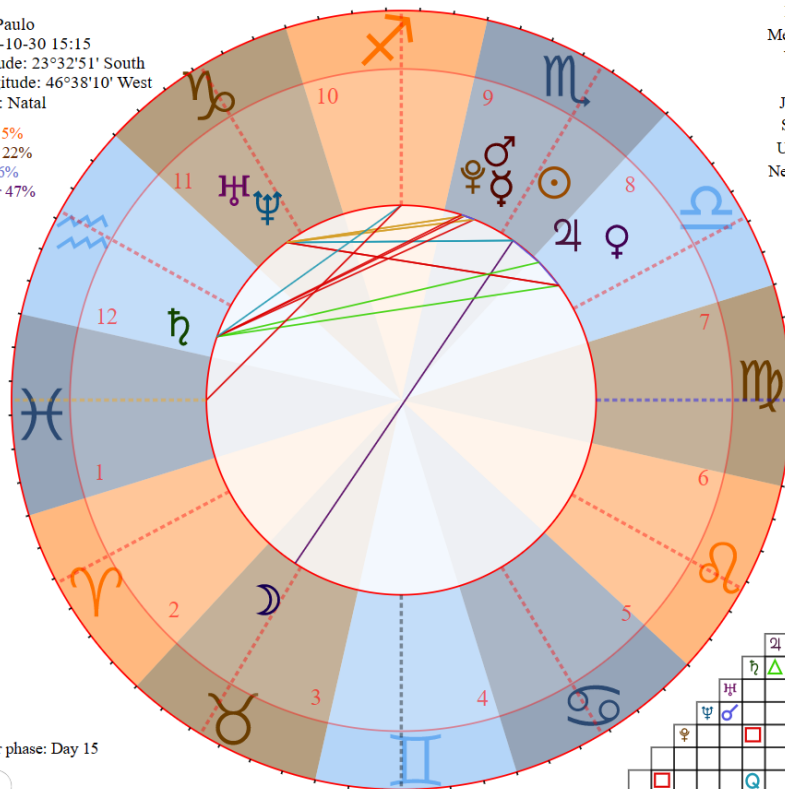
Exemplo:

## Luiza

Info:  
São Paulo  
1993-10-30 15:15  
Latitude: 23°32'51" South  
Longitude: 46°38'10" West  
Type: Natal

Fire 15%  
Earth 22%  
Air 16%  
Water 47%

Lunar phase: Day 15



### Planets and houses for Luiza:

Sun	♈	07°17'58" M	Cusp 1:	♈	12°45'44"
Moon	♒	09°24'15" S	Cusp 2:	♉	10°46'26"
Mercury	♑	20°50'49" M R	Cusp 3:	♊	11°18'17"
Venus	♒	18°17'27" S	Cusp 4:	♋	12°36'10"
Mars	♑	23°13'44" M	Cusp 5:	♌	13°29'32"
Jupiter	♐	27°43'17" S	Cusp 6:	♍	13°41'29"
Saturn	♒	23°38'21" S	Cusp 7:	♎	12°45'44"
Uranus	♒	18°41'35" S	Cusp 8:	♏	10°46'26"
Neptune	♒	18°38'01" S	Cusp 9:	♐	11°18'17"
Pluto	♒	24°42'45" M	Cusp 10:	♑	12°36'10"
Asc		12°45'44" M	Cusp 11:	♒	13°29'32"
Mc		12°36'10" S	Cusp 12:	♓	13°41'29"



Precisamos quebrar as informações dessa imagem sendo:

- Mandala, círculo com os planetas;
- As informações do "Planets and houses for Luiza";
- O gráfico abaixo com os aspectos;