



Pose Frame Specification for SDFormat 1.7

Addisu Taddese and Steve Peters (Open Robotics)

Eric Cousineau (TRI)

Outline

What is SDFormat?

Proposed capabilities

Existing pose frame semantics in SDFormat 1.6

Pose frame semantics in SDFormat 1.7

Target release dates

Demos

What is SDFormat?

Simulation Description Format (SDFormat) is an XML format for describing objects and environments for robot simulators.

Similar to URDF, but:

- Can describe the *world* (gravity, wind, lights, etc.), not just a robot
- Handles schema versions
- Admits more topology (e.g. closed loops)
- Motivated by Gazebo

Proposed capabilities for SDFormat 1.7

Specifying relative poses

```
<pose relative_to="frame_name">0 0 1 0 0 0</pose>
```

Implicit frames for links, joints, and models

Create named frames

```
<frame name="frame_name" attached_to="link_name" />
```

Notation

From the [Specifying pose](#) tutorial:

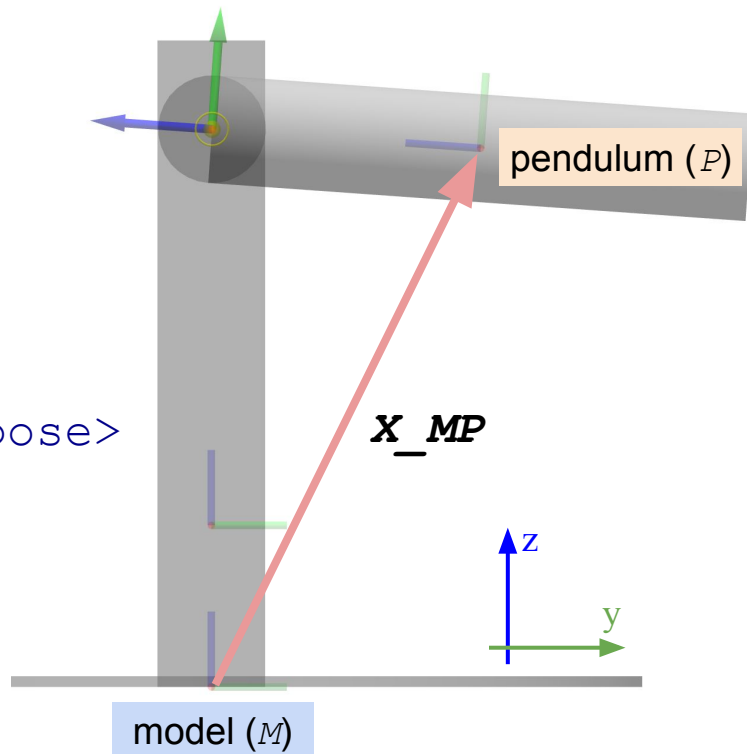
X_{MP} - pose of pendulum frame P
relative to model frame M

SDFormat's <pose> tag

```
<pose>0 0.5 1.0 1.57 0 0</pose>
```

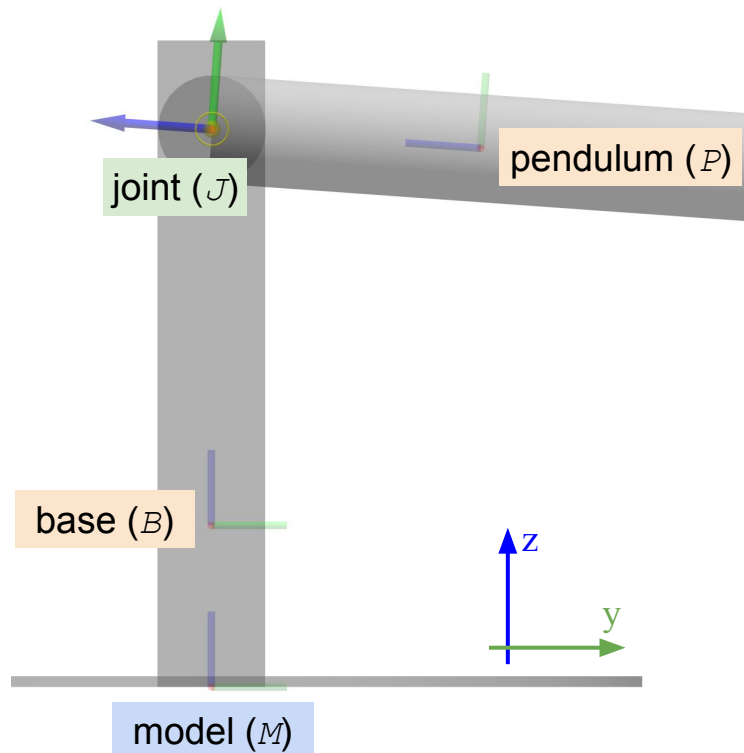
$\underbrace{\hspace{10em}}$ $\underbrace{\hspace{10em}}$

$x \ y \ z$ $r \ p \ y$



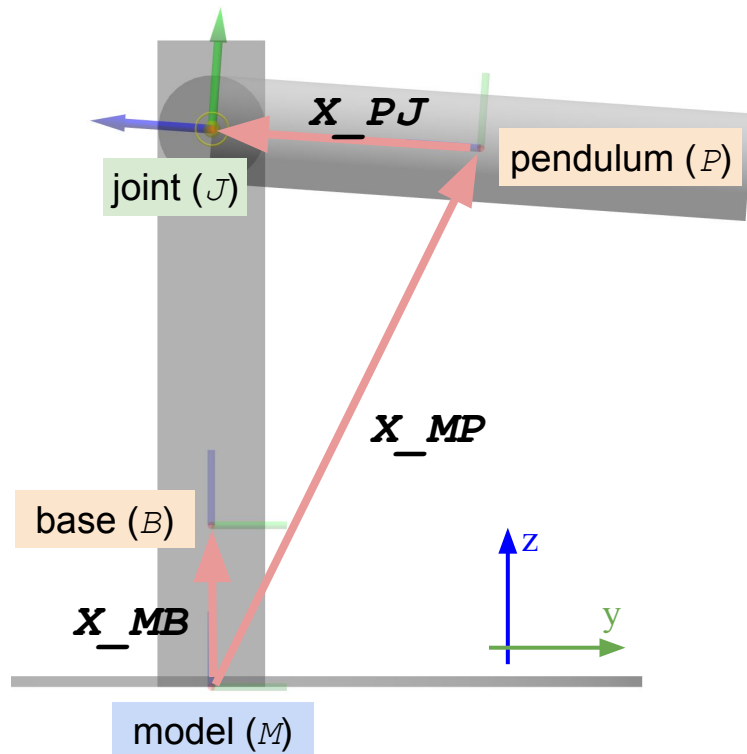
Pose frame semantics in SDFFormat 1.6

```
<sdf version="1.6">
  <model name="pendulum_with_base">
    <link name="base">
      <pose>0 0 0.3 0 0 0</pose>
    </link>
    <link name="pendulum">
      <pose>0 0.5 1.0 1.57 0 0</pose>
    </link>
    <joint name="joint" type="revolute">
      <parent>base</parent>
      <child>pendulum</child>
      <pose>0 0 0.5 0 0 0</pose>
      <axis>
        <xyz>1 0 0</xyz>
      </axis>
    </joint>
  </model>
</sdf>
```



Pose frame semantics in SDFFormat 1.6

```
<sdf version="1.6">
M <model name="pendulum_with_base">
B   <link name="base">
X_MB   <pose>0 0 0.3  0 0 0</pose>
        </link>
P   <link name="pendulum">
X_MP   <pose>0 0.5 1.0  1.57 0 0</pose>
        </link>
J   <joint name="joint" type="revolute">
        <parent>base</parent>
        <child>pendulum</child>
X_PJ   <pose>0 0 0.5  0 0 0</pose>
        <axis>
          <xyz>1 0 0</xyz>
        </axis>
        </joint>
      </model>
</sdf>
```

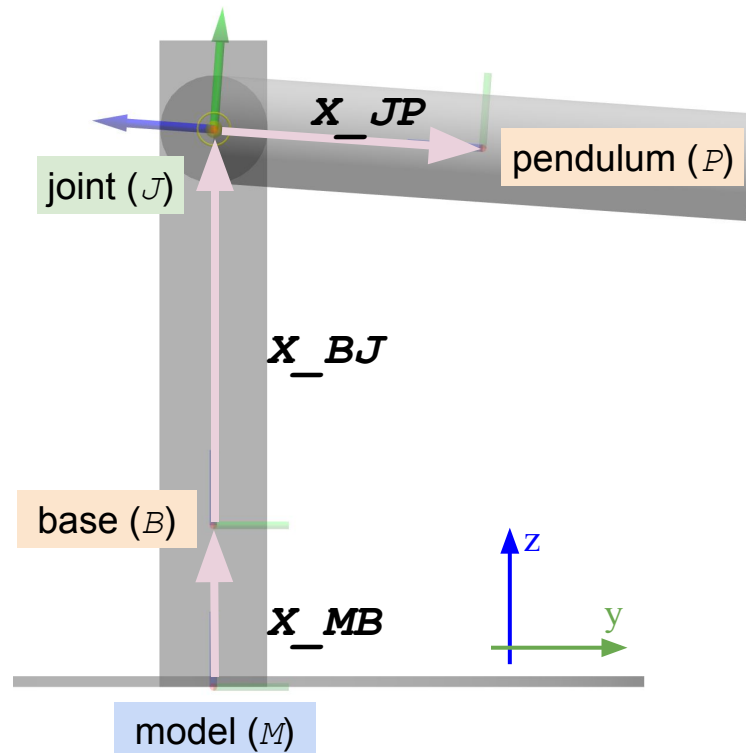


Pose frame semantics in SDFFormat 1.7

```

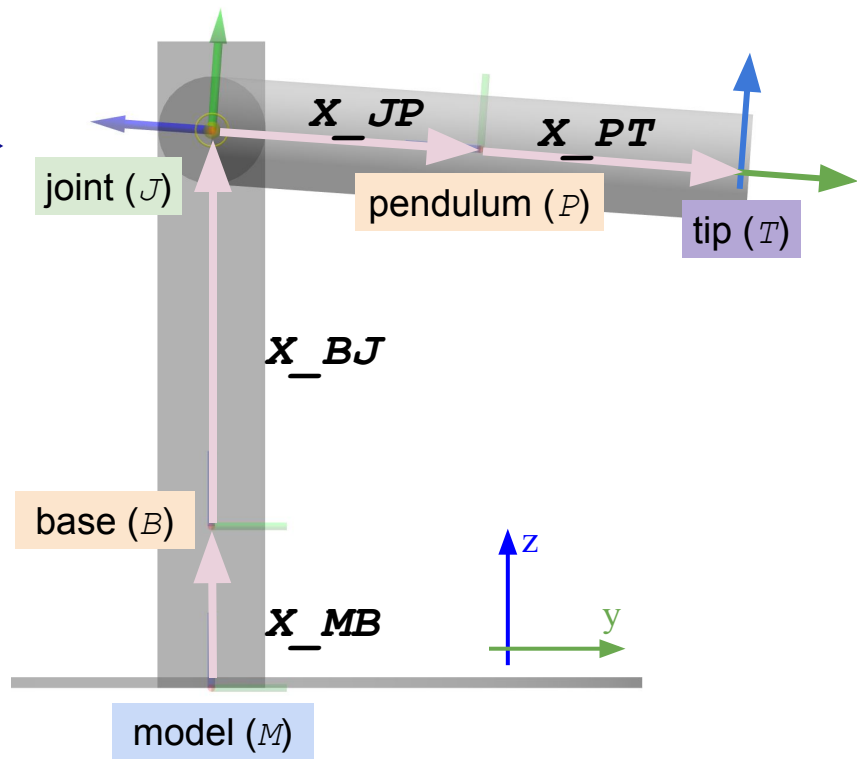
<sdf version="1.7">
M <model name="pendulum_with_base">
  B  <link name="base">
X_MB  <pose>0 0 0.3  0 0 0</pose>
    </link>
  P  <link name="pendulum">
    <pose relative_to="joint">
X_JP  0 0 -0.5  0 0 0
    </pose>
    </link>
  J  <joint name="joint" type="revolute">
    <parent>base</parent>
    <child>pendulum</child>
    <pose relative_to="base">
X_BJ  0 0 0.73  1.57 0 0
    </pose>
    <axis>
      <xyz>1 0 0</xyz>
    </axis>
    </joint>
  </model>
</sdf>

```



Named frames in SDFFormat 1.7

```
<sdf version="1.6">
  <model name="pendulum_with_base">
    ...
    T   <frame name="tip" attached_to="pendulum">
X_PT   <pose>0 0 -0.5 -1.57079 0 0</pose>
        </frame>
    P   <link name="pendulum">
V       <visual name="tip_visual">
X_TV    <pose relative_to="tip"/>
        ...
      </visual>
    </link>
  </model>
</sdf>
```



Additional Details

- Canonical links
- Naming requirements for links, joints, frames, etc.
- Cycle checks for frames
- `@expressed_in` instead of `use_parent_model_frame`

Read more in [Pose Frame Semantics Proposal](#)

Target release dates for SDFormat 1.7

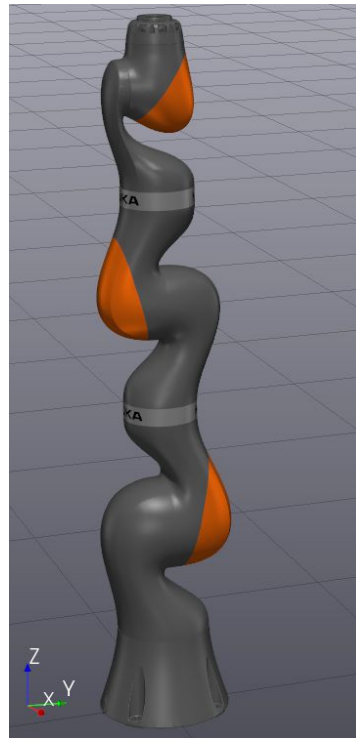
libsdformat: Dec. 15, 2019 (possibly earlier)

Gazebo (Classic): Jan. 2020 (Gazebo 11 release)

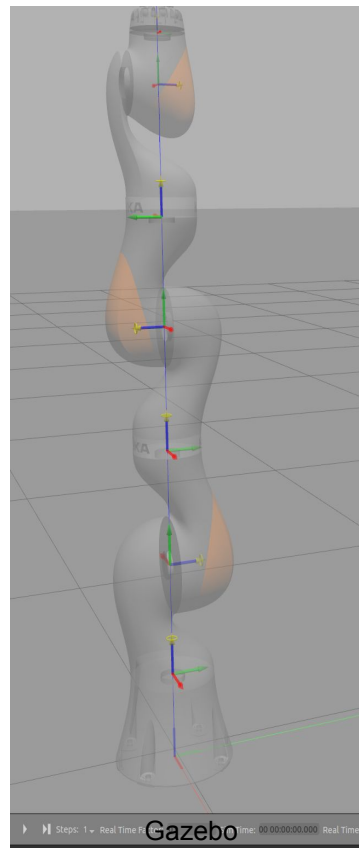
Drake: Dec. 15, 2019 (possibly using libsdformat prerelease)

Demos / Examples

- Convert URDF to SDFFormat 1.7
- Demo in Drake: KUKA IIWA
- Demo in Gazebo: KUKA IIWA



Drake
[\[dev pr\]](#) | [\[commit\]](#)



Gazebo
[\[dev pr\]](#) | [\[commit\]](#)

Loaded using SDFFormat 1.7, libsdformat8 (dev branch)

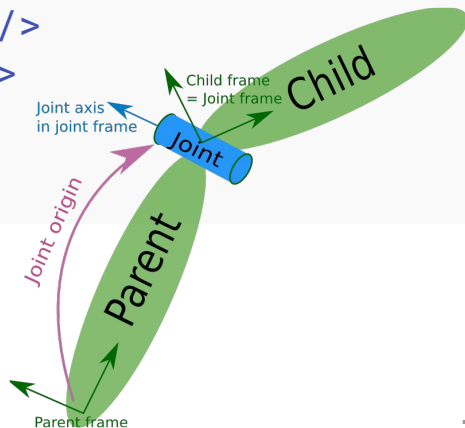
Example: Parity with URDF

URDF

```
<link name="L1"/>

<joint name="joint1"
      type="revolute">
  <origin xyz="0.1 0.2 0.3"
        rpy="0.4 0.5 0.6"/>
  <parent link="L1"/>
  <child link="L2"/>
</joint>

<link name="L2"/>
```

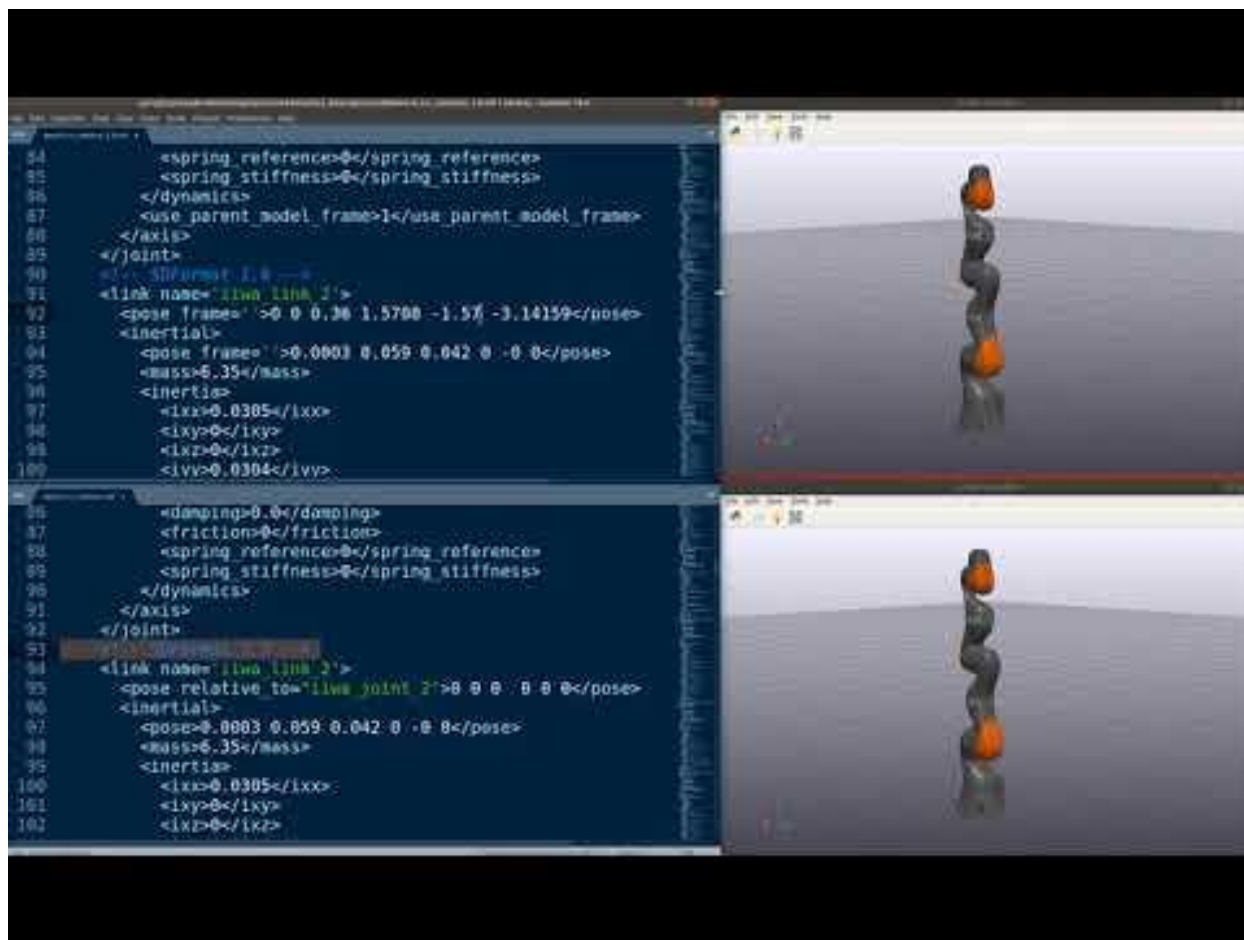


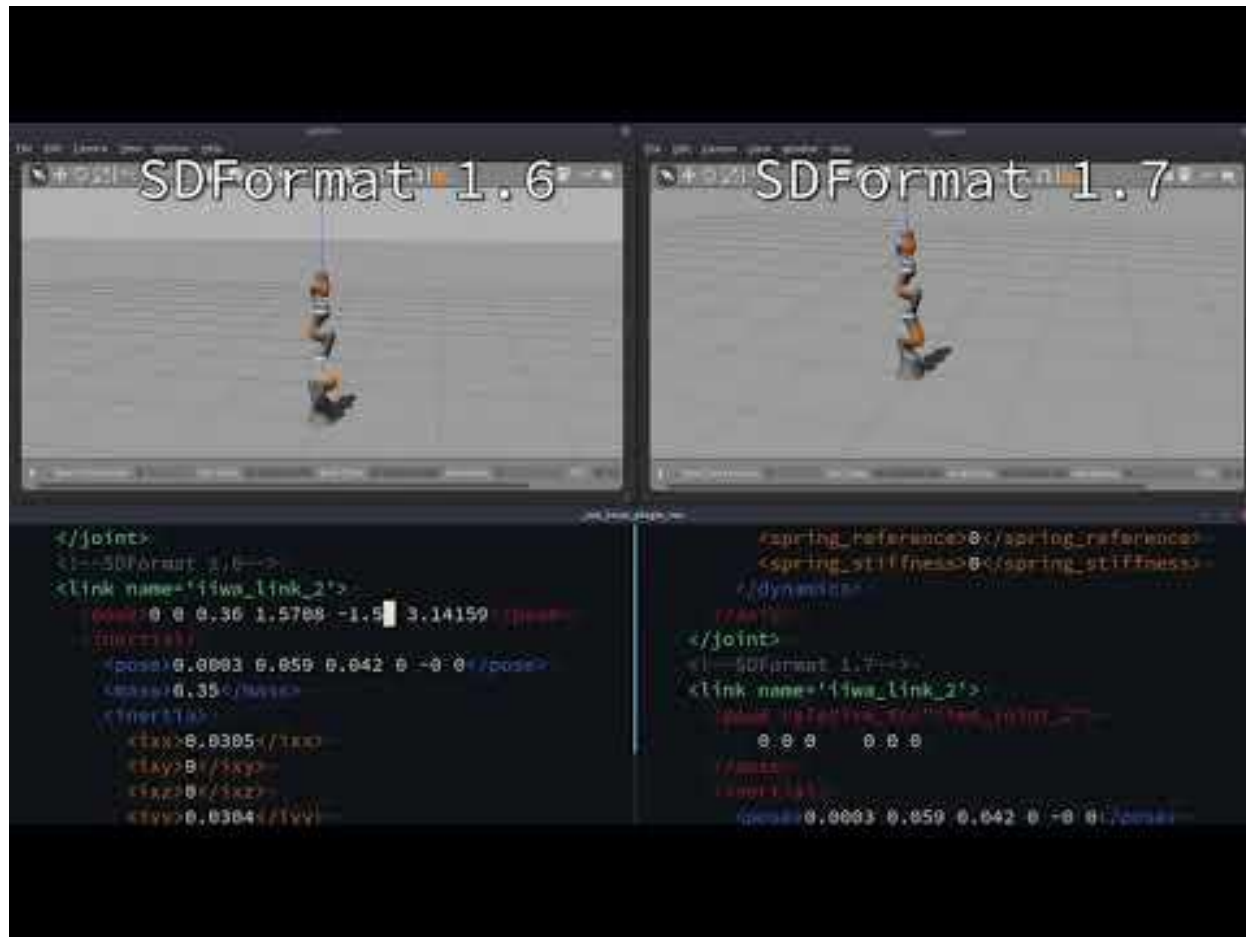
SDFFormat 1.7

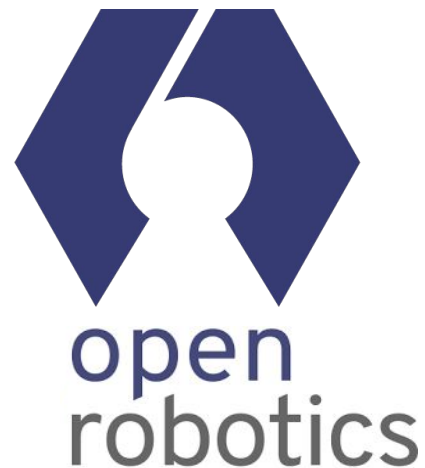
```
<link name="L1"/>

<joint name="joint1"
      type="revolute">
  <pose relative_to="L1">
    0.1 0.2 0.3    0.4 0.5 0.6
  </pose>
  <parent>L1</parent>
  <child>L2</child>
</joint>

<link name="L2">
  <pose relative_to="joint1" />
</link>
```







Thank You