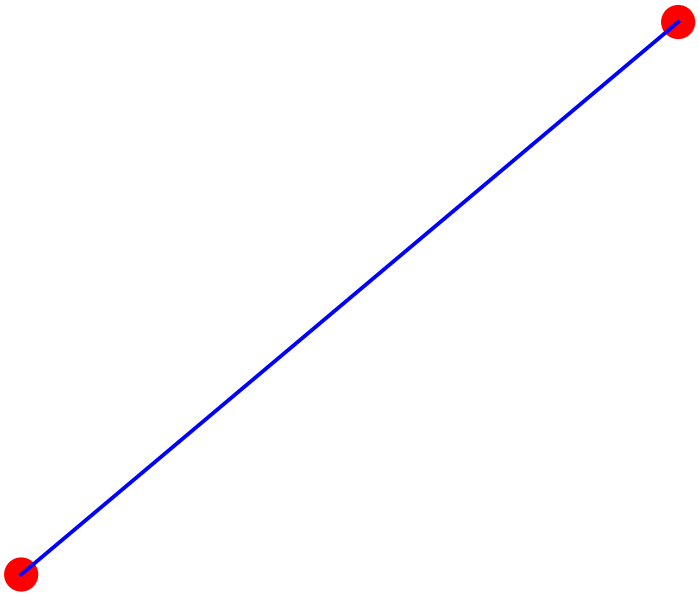
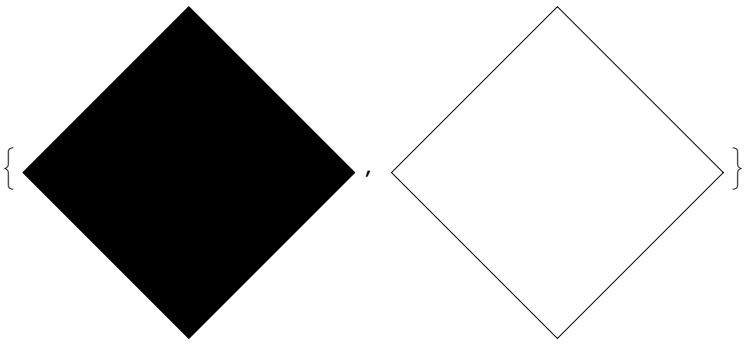
```
(*  Graphics Complexes:  WQ,  FS 2013 *)


gc = GraphicsComplex[{{1.2, 3.5}, {5.6, 7.2}},
   {{Red, Point[1], Point[2]}, Line[{1, 2}]}]
GraphicsComplex[{{1.2, 3.5}, {5.6, 7.2}},
 {{RGBColor[1, 0, 0], Point[1], Point[2]}, Line[{1, 2}]}]

Graphics[{Blue, PointSize[0.05], Thick, gc}]
```

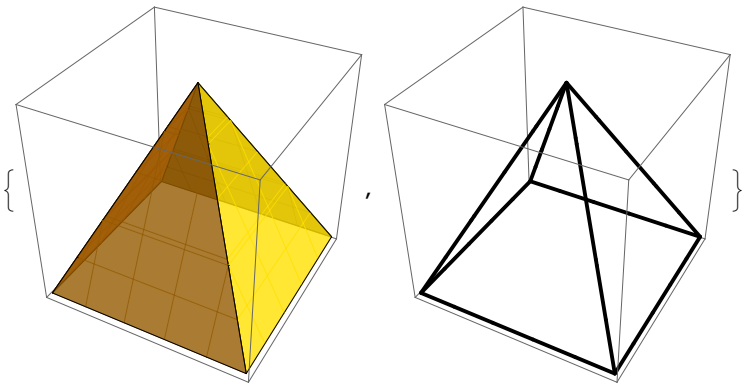


```
v = {{1, 0}, {0, 1}, {-1, 0}, {0, -1}};

{Graphics[GraphicsComplex[v, Polygon[{1, 2, 3, 4}]]],
 Graphics[GraphicsComplex[v, Line[{1, 2, 3, 4, 1}]]]}
```



```
v = {{0, 0, 0}, {2, 0, 0}, {2, 2, 0}, {0, 2, 0}, {1, 1, 2}};
i = {{1, 2, 5}, {2, 3, 5}, {3, 4, 5}, {4, 1, 5}};
```

```
{Graphics3D[{Opacity[.8], Yellow, GraphicsComplex[v, Polygon[i]]}],
 Graphics3D[{Thick, GraphicsComplex[v, Line[i]]}]}
```

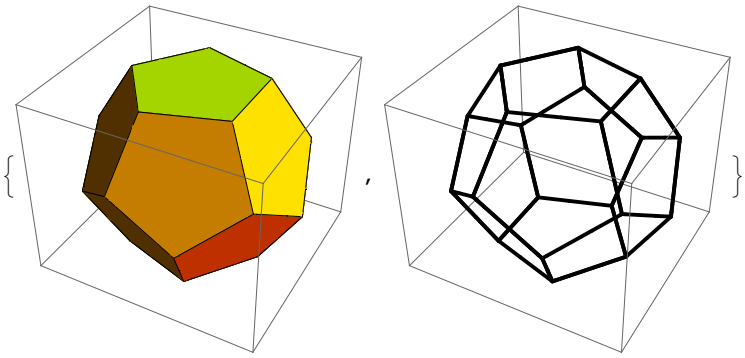


Use built - in PolyhedronData :

```
v = PolyhedronData["Dodecahedron", "VertexCoordinates"];
Short[i = PolyhedronData["Dodecahedron", "FaceIndices"]]
```

```
{{15, 10, 9, 14, 1}, {2, 6, 12, 11, 5}, <<8>>, {3, 7, 16, 1, 14}, {16, 8, 4, 15, 1}}
```

```
{Graphics3D[{Yellow, GraphicsComplex[v, Polygon[i]]}],
 Graphics3D[{Thick, GraphicsComplex[v, Line[i]]}]}
```



```
PolyhedronData[]
```

```
platoKoerp = PolyhedronData["Platonic"]
```

```
{Cube, Dodecahedron, Icosahedron, Octahedron, Tetrahedron}
```

```
PolyhedronData["Cube", "Properties"]
```

```
{AdjacentFaceIndices, AlternateNames, AlternateStandardNames, Amphichiral,
 Antiprism, Archimedean, ArchimedeanDual, Centroid, Chiral, Circumcenter,
 Circumradius, Circumsphere, Classes, Compound, Concave, Convex, Cuboid,
 DefaultOrientation, Deltahedron, DihedralAngleRules, DihedralAngles,
 Dipyramid, DualCompound, DualName, DualScale, EdgeCount, EdgeIndices,
 EdgeLengths, Edges, Equilateral, FaceCount, FaceCountRules, FaceIndices,
 Faces, GeneralizedDiameter, Hypercube, Image, Incenter, InertiaTensor,
 Information, Inradius, Insphere, Isohedron, Johnson, KeplerPoinsot, Midcenter,
 Midradius, Midsphere, MultipieceNetCoordinates, MultipieceNetFaceIndices,
 MultipieceNetImage, Name, NetCoordinates, NetCount, NetEdgeIndices, NetEdges,
 NetFaceIndices, NetFaces, NetGraph, NetImage, NotationRules, Orientations,
 Orthotope, Platonic, PlatonicDual, PolyhedronIndices, Prism, Pyramid,
 Quasiregular, RectangularParallelepiped, RegionFunction, Rhombohedron,
 Rigid, SchlaefliSymbol, SelfDual, Shaky, Simplex, SkeletonCoordinates,
 SkeletonGraph, SkeletonGraphName, SkeletonImage, SkeletonRules, SpaceFilling,
 StandardName, StandardNames, Stellation, StellationCount, SurfaceArea,
 SymmetryGroupString, Uniform, UniformDual, VertexCoordinates, VertexCount,
 VertexIndices, VertexSubsetHulls, Volume, WythoffSymbol, Zonohedron}
```

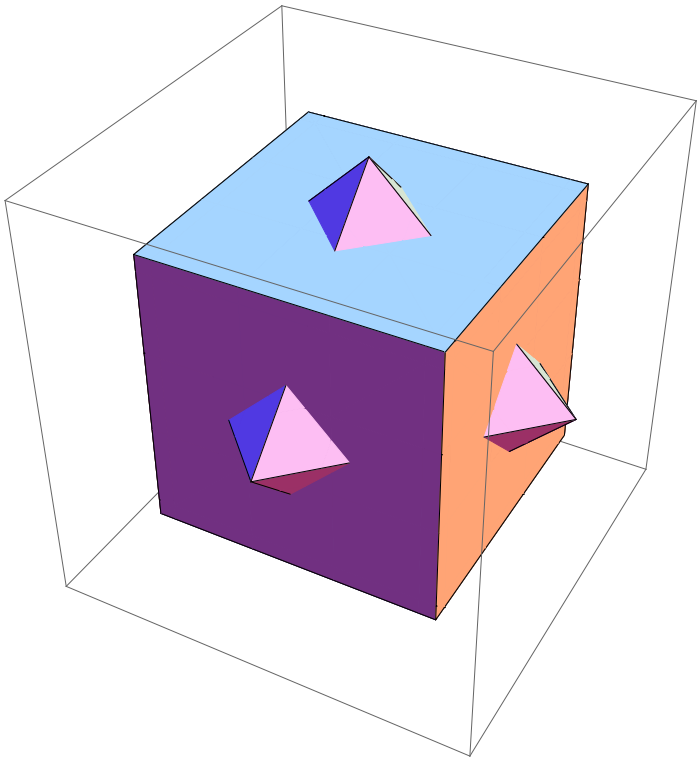
```
cube = PolyhedronData["Cube", "Faces"]
```

```
GraphicsComplex[{{-1/2, -1/2, -1/2}, {-1/2, -1/2, 1/2}, {-1/2, 1/2, -1/2}, {-1/2, 1/2, 1/2},
  {1/2, -1/2, -1/2}, {1/2, -1/2, 1/2}, {1/2, 1/2, -1/2}, {1/2, 1/2, 1/2}}, Polygon[
  {{8, 4, 2, 6}, {8, 6, 5, 7}, {8, 7, 3, 4}, {4, 3, 1, 2}, {1, 3, 7, 5}, {2, 1, 5, 6}}]]
```

```
octa = PolyhedronData["Octahedron", "Faces"]
```

```
GraphicsComplex[{{-1/Sqrt[2], 0, 0}, {0, 1/Sqrt[2], 0}, {0, 0, -1/Sqrt[2]},
  {0, 0, 1/Sqrt[2]}, {0, -1/Sqrt[2], 0}, {1/Sqrt[2], 0, 0}}, Polygon[{{4, 5, 6}, {4, 6, 2},
   {4, 2, 1}, {4, 1, 5}, {5, 1, 3}, {5, 3, 6}, {3, 1, 2}, {6, 3, 2}}]]
```

```
Graphics3D[{cube, octa}]
```



```
radii = {#, PolyhedronData[#, "Inradius"], PolyhedronData[#, "Circumradius"]} & /@
    platoKoerp;
Text@Grid[Prepend[radii, {"Körper", "Inkugelradius", "Umkugelradius"}],
  Dividers → {{{True}}, {True, True, {False}, True}}, Spacings → {1, 2}]
```

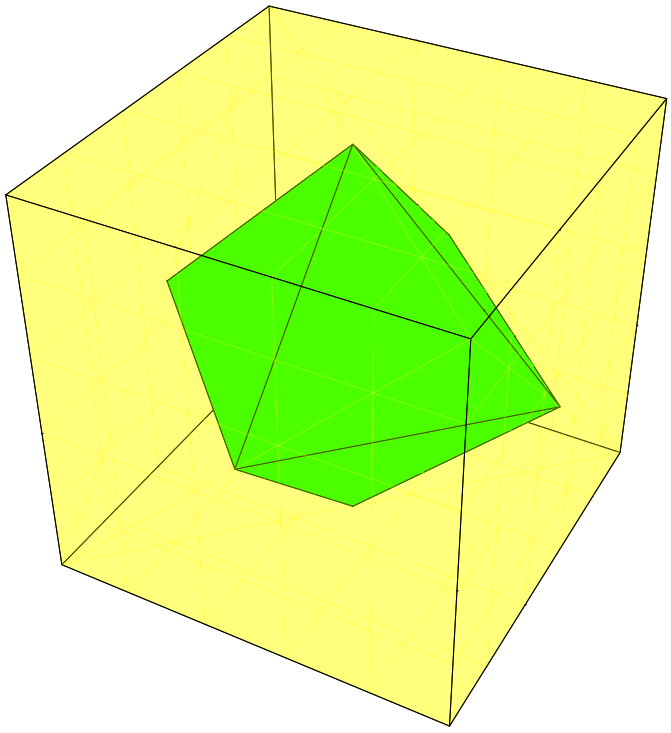
Körper	Inkugelradius	Umkugelradius
Cube	$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$
Dodecahedron	$\frac{1}{20}\sqrt{250+110\sqrt{5}}$	$\frac{1}{4}\left(\sqrt{3}+\sqrt{15}\right)$
Icosahedron	$\frac{1}{12}\left(3\sqrt{3}+\sqrt{15}\right)$	$\frac{1}{4}\sqrt{10+2\sqrt{5}}$
Octahedron	$\frac{1}{\sqrt{6}}$	$\frac{1}{\sqrt{2}}$
Tetrahedron	$\frac{1}{2\sqrt{6}}$	$\frac{\sqrt{\frac{3}{2}}}{2}$

```
(* Setze verkleinertes Octahedron in durchsichtigen Würfel *)

trans[t_, g_GraphicsComplex] :=
 GraphicsComplex[(DiagonalMatrix[{t, t, t}].# &) /@ g[[1]], g[[2]] ]
```

```
trans[1 / Sqrt[2], octa]
```

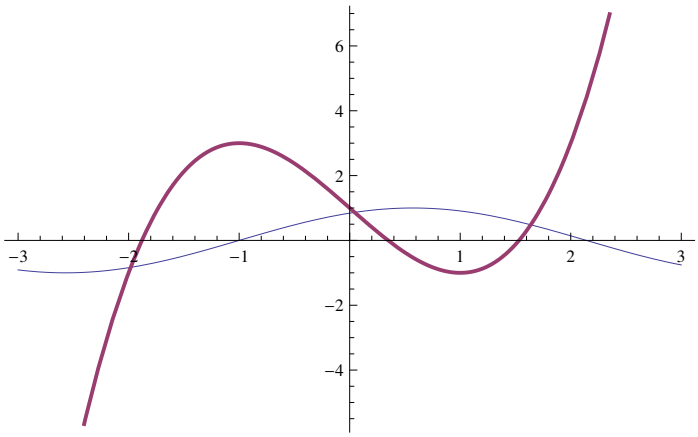
```
GraphicsComplex[
 {{-1/2, 0, 0}, {0, 1/2, 0}, {0, 0, -1/2}, {0, 0, 1/2}, {0, -1/2, 0}, {1/2, 0, 0}},
 Polygon[{{4, 5, 6}, {4, 6, 2}, {4, 2, 1},
   {4, 1, 5}, {5, 1, 3}, {5, 3, 6}, {3, 1, 2}, {6, 3, 2}}]]
```

```
Graphics3D[{{Yellow, Opacity[0.3], cube},
  {Green, trans[1 / Sqrt[2], octa]}},
 PlotRange → All, Boxed → False, Lighting → {White}]
```



```
(* Verbinde Graphics und Numerik: Flaeche zwischen 2 Kurven *)
```

```
f = Sin[x + 1];
g = x^3 - 3 x + 1;
zweiKurven = Plot[{f, g}, {x, -3, 3}, PlotStyle → {{}, Thick}]
```




```
x1 = x /. FindRoot[f == g, {x, -2}]
x2 = x /. FindRoot[f == g, {x, 0}]
x3 = x /. FindRoot[f == g, {x, 2}]
```

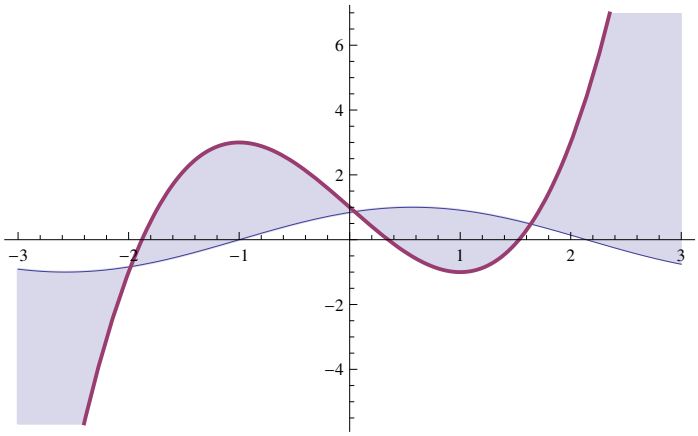
```
-1.98099
```

```
0.0450477
```

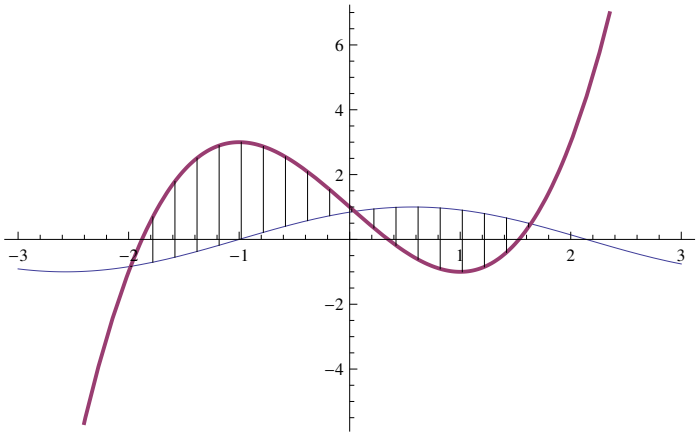
```
1.6383
```

```
(* Wir wollen das Gebiet straffieren *)
```

```
Plot[{f, g}, {x, -3, 3}, PlotStyle → {{}, Thick}, Filling → {1 → {2}}]
```



```
(* mit selbstgestrickter Schraffur *)
Show[zweiKurven, Table[Graphics[Line[{{x, f}, {x, g}}]], {x, x1, x3, 0.2}]]
```




```
(* Nun die Integration *)

links = Integrate[1, {x, x1, x2}, {y, f, g}]
```

```
4.00506
```

```
rechts = Integrate[1, {x, x2, x3}, {y, g, f}]
```

```
2.0066
```

```
flaeche = rechts + links
```

```
6.01167
```

```
(* Kontrolle *)

NIntegrate [Abs[f - g], {x, x1, x3}]
```

```
6.01167
```