```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {
    "colab_type": "text",
    "id": "mMcWEbFiLKvn"
   },
   "source": [
    "# NumPy Exercises \n",
    "\n",
    "Now that we've learned about NumPy let's test your knowledge. We'll start off with a few simple tasks, and then you'll be asked some more complicated questions."
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "colab_type": "text",
    "id": "DvBjXHM1LKvw"
   },
   "source": [
    "#### Import NumPy as np"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 1,
   "metadata": {
    "colab": {},
    "colab_type": "code",
    "id": "StInvTamLKv0"
   },
   "outputs": [],
   "source": [
    "import numpy as np"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "colab_type": "text",
    "id": "qav3VCgyLKv5"
   },
   "source": [
    "#### Create an array of 10 zeros "
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "metadata": {
    "colab": {},
    "colab_type": "code",
    "id": "gLlF23CdLKv6",
    "outputId": "824bc0c2-05e8-4a2f-b02c-803eb133cb1e"
   },
```

```
 "outputs": [],
 "source": [
  "z = np.zeros(10)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "colab_type": "text",
  "id": "-r7m8k4vLKv_"
 },
 "source": [
  "#### Create an array of 10 ones"
 ]
},
{
 "cell_type": "code",
 "execution_count": 3,
 "metadata": {
  "colab": {},
  "colab_type": "code",
  "id": "iwuCBhvcLKwB",
  "outputId": "2358e05b-b357-4b0e-d4c4-fefc91889469"
 },
 "outputs": [],
 "source": [
  "a = np.ones(10)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "colab_type": "text",
  "id": "FIcddv6lLKwO"
 },
 "source": [
  "#### Create an array of 10 fives"
 ]
},
{
 "cell_type": "code",
 "execution_count": 4,
 "metadata": {
  "colab": {},
  "colab_type": "code",
  "id": "tXDJ5b8cLKwU",
  "outputId": "f784a7d9-a26d-4d97-eeee-15e78e84b014"
 },
 "outputs": [],
 "source": [
  "b = np.ones(10)*5"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "colab_type": "text",
  "id": "4UInvEwQLKwg"
```

```
 },
 "source": [
  "#### Create an array of the integers from 10 to 50"
 ]
},
{
 "cell_type": "code",
 "execution_count": 8,
 "metadata": {
  "colab": {},
  "colab_type": "code",
  "id": "Z9kAbHHiLKwj",
  "outputId": "1b2d4a50-b3c4-44af-dd3b-643a6546d019"
 },
 "outputs": [],
 "source": [
  "c = np.arange(10,51)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "colab_type": "text",
  "id": "3ZUIzdDeLKww"
 },
 "source": [
  "#### Create an array of all the even integers from 10 to 50"
 ]
},
{
 "cell_type": "code",
 "execution_count": 9,
 "metadata": {
  "colab": {},
  "colab_type": "code",
  "id": "dVVfz73BLKw5",
  "outputId": "ee6de97d-f84f-4f4b-d6d6-c21c46f59666"
 },
 "outputs": [],
 "source": [
  "d = np.arange(10,51,2)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "colab_type": "text",
  "id": "6e200i5-LKxB"
 },
 "source": [
  "#### Create a 3x3 matrix with values ranging from 0 to 8"
 ]
},
{
 "cell_type": "code",
 "execution_count": 10,
 "metadata": {
  "colab": {},
```

```
  "colab_type": "code",
  "id": "6ODYRVCxLKxF",
  "outputId": "0327b8fd-1f75-4da0-ddbc-9193a7cbbd3d"
 },
 "outputs": [],
 "source": [
  "e = np.array([[0,1,2],[3,4,5],[6,7,8]])"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "colab_type": "text",
  "id": "54wb4NboLKxN"
 },
 "source": [
  "#### Create a 3x3 identity matrix"
 ]
},
{
 "cell_type": "code",
 "execution_count": 11,
 "metadata": {
  "colab": {},
  "colab_type": "code",
  "id": "j8JQg-VSLKxT",
  "outputId": "697e3eed-3266-4f57-90ab-ff5d5c430c7d"
 },
 "outputs": [],
 "source": [
  "e = np.array([[1,0,0],[0,1,0],[0,0,1]])"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "colab_type": "text",
  "id": "_bpkEUARLKxa"
 },
 "source": [
  "#### Use NumPy to generate a random number between 0 and 1"
 ]
},
{
 "cell_type": "code",
 "execution_count": 12,
 "metadata": {
  "colab": {},
  "colab_type": "code",
  "id": "Co4I1qACLKxc",
  "outputId": "efdd03f8-868e-4eeb-e40f-e75dba4cd14e"
 },
 "outputs": [],
 "source": [
  "f = np.random.random()"
 ]
},
{
```

```
   "cell_type": "markdown",
   "metadata": {
    "colab_type": "text",
    "id": "7Qix8s61LKxp"
   },
   "source": [
    "#### Use NumPy to generate an array of 25 random numbers sampled
from a standard normal distribution"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 13,
   "metadata": {
    "colab": {},
    "colab_type": "code",
    "id": "eUgQBgpQLKxs",
    "outputId": "04664997-3ff0-421e-b92e-5e12c80321f9"
   },
   "outputs": [],
   "source": [
    "g = np.random.normal(size=25)"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "colab_type": "text",
    "id": "-sL9HYlWLKx1"
   },
   "source": [
    "#### Create the following matrix:"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 15,
   "metadata": {
    "colab": {},
    "colab_type": "code",
    "id": "p2pm5Mm6LKx5",
    "outputId": "aa76f1ed-5967-4d3d-8061-72f82595c599"
   },
   "outputs": [],
   "source": [
    "h = np.arange(0.01, 1.01, 0.01).reshape(10, 10)"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "colab_type": "text",
    "id": "UQzj3r7NLKx_"
   },
   "source": [
    "#### Create an array of 20 linearly spaced points between 0 and 1:"
   ]
  },
  },
```

```
  {
   "cell_type": "code",
   "execution_count": 16,
   "metadata": {
    "colab": {},
    "colab_type": "code",
    "id": "AniTWWz9LKyC",
    "outputId": "dfb092f8-2f1e-446a-99e2-5c6df71631ac"
   },
   "outputs": [],
   "source": [
    "i = np.arange(0,1,0.05).reshape(4,5)"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "colab_type": "text",
    "id": "i3vsM3rPLKyK"
   },
   "source": [
    "## Numpy Indexing and Selection\n",
    "\n",
    "Now you will be given a few matrices, and be asked to replicate the
resulting matrix outputs:"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 17,
   "metadata": {
    "colab": {},
    "colab_type": "code",
    "id": "AM_s5FAvLKyN",
    "outputId": "b29d6933-31ca-4d45-d4c9-123877d1e7be"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "array([[ 1,  2,  3,  4,  5],\n",
       "       [ 6,  7,  8,  9, 10],\n",
       "       [11, 12, 13, 14, 15],\n",
       "       [16, 17, 18, 19, 20],\n",
       "       [21, 22, 23, 24, 25]])"
      ]
     },
     "execution_count": 17,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "mat = np.arange(1,26).reshape(5,5)\n",
    "mat"
   ]
  },
  {
```

```
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "colab": {},
  "colab_type": "code",
  "id": "rxlUOQ4dLKyX"
 },
 "outputs": [],
 "source": [
  "# WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW\n",
  "# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T\n",
  "# BE ABLE TO SEE THE OUTPUT ANY MORE"
 ]
},
{
 "cell_type": "code",
 "execution_count": 18,
 "metadata": {
  "colab": {},
  "colab_type": "code",
  "id": "2BJ3cylOLKyf",
  "outputId": "ab55bbee-37f2-4797-bad5-4ed340f4d684"
 },
 "outputs": [
  {
   "data": {
    "text/plain": [
     "array([[12, 13, 14, 15],\n",
     "       [17, 18, 19, 20],\n",
     "       [22, 23, 24, 25]])"
    ]
   },
   "execution_count": 18,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "mat[2:, 1:]\n"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "colab": {},
  "colab_type": "code",
  "id": "9-U53vNoLKyw"
 },
 "outputs": [],
 "source": [
  "# WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW\n",
  "# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T\n",
  "# BE ABLE TO SEE THE OUTPUT ANY MORE"
 ]
},
{
 "cell_type": "code",
```

```
 "execution_count": 26,
 "metadata": {
  "colab": {},
  "colab_type": "code",
  "id": "QJyclCVfLKy6",
  "outputId": "9762e12a-13fe-4567-fd7a-718402be1edb"
 },
 "outputs": [
  {
   "data": {
    "text/plain": [
     "20"
    ]
   },
   "execution_count": 26,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "mat[3,4]"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "colab": {},
  "colab_type": "code",
  "id": "E_PuvjL5LKzH"
 },
 "outputs": [],
 "source": [
  "# WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW\n",
  "# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T\n",
  "# BE ABLE TO SEE THE OUTPUT ANY MORE"
 ]
},
{
 "cell_type": "code",
 "execution_count": 30,
 "metadata": {
  "colab": {},
  "colab_type": "code",
  "id": "TzFTd7eHLKzY",
  "outputId": "5aa2b71d-7d91-4bbd-8550-46aa7e62385d"
 },
 "outputs": [
  {
   "data": {
    "text/plain": [
     "array([[ 2],\n",
     "       [ 7],\n",
     "       [12]])"
    ]
   },
   "execution_count": 30,
   "metadata": {},
```

```
   "output_type": "execute_result"
  }
 ],
 "source": [
  "mat[:3, 1:2]"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "colab": {},
  "colab_type": "code",
  "id": "BaybacXxLKze"
 },
 "outputs": [],
 "source": [
  "# WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW\n",
  "# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T\n",
  "# BE ABLE TO SEE THE OUTPUT ANY MORE"
 ]
},
{
 "cell_type": "code",
 "execution_count": 32,
 "metadata": {
  "colab": {},
  "colab_type": "code",
  "id": "hvblpMEFLKzl",
  "outputId": "95ecd0ff-f24a-4e5f-edae-481966a15a0a"
 },
 "outputs": [
  {
   "data": {
    "text/plain": [
     "array([[21, 22, 23, 24, 25]])"
    ]
   },
   "execution_count": 32,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "mat[4:]"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "colab": {},
  "colab_type": "code",
  "id": "L0SoIZfILKzr"
 },
 "outputs": [],
 "source": [
  "# WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW\n",
```

```
   "# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T\n",
   "# BE ABLE TO SEE THE OUTPUT ANY MORE"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 38,
  "metadata": {
   "colab": {},
   "colab_type": "code",
   "id": "h2NmY81aLKzv",
   "outputId": "ec3006aa-5bd4-43ef-b876-f0475c9a26ce"
  },
  "outputs": [
   {
    "data": {
     "text/plain": [
      "array([[16, 17, 18, 19, 20],\n",
      "       [21, 22, 23, 24, 25]])"
     ]
    },
    "execution_count": 38,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "mat[3:5]"
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {
   "colab_type": "text",
   "id": "40ijP69QLKz6"
  },
  "source": [
   "### Now do the following"
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {
   "colab_type": "text",
   "id": "hJYmPKZ6LKz7"
  },
  "source": [
   "#### Get the sum of all the values in mat"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 39,
  "metadata": {
   "colab": {},
   "colab_type": "code",
   "id": "lgjWcsNHLKz9",
   "outputId": "a4aebb34-800e-490a-94fa-b2f29e05ec8a"
```

```
 },
 "outputs": [],
 "source": [
  "sum = np.sum(mat)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "colab_type": "text",
  "id": "aOxAVLe8LK0B"
 },
 "source": [
  "#### Get the standard deviation of the values in mat"
 ]
},
{
 "cell_type": "code",
 "execution_count": 40,
 "metadata": {
  "colab": {},
  "colab_type": "code",
  "id": "1bExi0tvLK0D",
  "outputId": "cdcf114d-bcd3-494a-afc8-f4b592566334"
 },
 "outputs": [],
 "source": [
  "sd = np.std(mat)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "colab_type": "text",
  "id": "qWvkrLQWLK0H"
 },
 "source": [
  "#### Get the sum of all the columns in mat"
 ]
},
{
 "cell_type": "code",
 "execution_count": 41,
 "metadata": {
  "colab": {},
  "colab_type": "code",
  "id": "-XFw4SD1LK0J",
  "outputId": "03cee652-bdb1-411c-a81f-7534525a3d21"
 },
 "outputs": [],
 "source": [
  "sum_of_columns = np.sum(mat, axis=0)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "colab_type": "text",
```

```
    "collapsed": true,
    "id": "x75g5hGlLK0N"
   },
   "source": []
  }
 ],
 "metadata": {
  "colab": {
   "name": "Numpy Exercise .ipynb",
   "provenance": []
  },
  "kernelspec": {
   "display_name": "Python 3",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.11.5"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 1
}
```