# week1-assignment.ipynb


```
{
 "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": [
    {
     "data": {
      "text/plain": [
       "array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])"
      ]
     },
     "execution_count": 2,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "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": [
    {
     "data": {
      "text/plain": [
       "array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])"
      ]
     },
     "execution_count": 3,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "np.ones(10)"
   ]
  },
```

```
  {
   "cell_type": "markdown",
   "metadata": {
    "colab_type": "text",
    "id": "FIcddv6lLKwO"
   },
   "source": [
    "#### Create an array of 10 fives"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 6,
   "metadata": {
    "colab": {},
    "colab_type": "code",
    "id": "tXDJ5b8cLKwU",
    "outputId": "f784a7d9-a26d-4d97-eeee-15e78e84b014"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])"
      ]
     },
     "execution_count": 6,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "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": 10,
   "metadata": {
    "colab": {},
    "colab_type": "code",
    "id": "Z9kAbHHiLKwj",
    "outputId": "1b2d4a50-b3c4-44af-dd3b-643a6546d019"
   },
   "outputs": [
    {
```

```
      "name": "stdout",
      "output_type": "stream",
      "text": [
       "[10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33\n",
       " 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50]\n"
      ]
     }
    ],
    "source": [
     "a=np.arange(10,51)\n",
     "print(a)"
    ]
   },
   {
    "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": 16,
    "metadata": {
     "colab": {},
     "colab_type": "code",
     "id": "dVVfz73BLKw5",
     "outputId": "ee6de97d-f84f-4f4b-d6d6-c21c46f59666"
    },
    "outputs": [
     {
      "name": "stdout",
      "output_type": "stream",
      "text": [
       "[10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50]\n"
      ]
     }
    ],
    "source": [
     "a1=np.arange(10,51,2)\n",
     "print(a1)"
    ]
   },
   {
    "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": 34,
   "metadata": {
    "colab": {},
    "colab_type": "code",
    "id": "6ODYRVCxLKxF",
    "outputId": "0327b8fd-1f75-4da0-ddbc-9193a7cbbd3d"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "array([[0, 1, 2],\n",
       "       [3, 4, 5],\n",
       "       [6, 7, 8]])"
      ]
     },
     "execution_count": 34,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "a2=np.arange(0,9).reshape(3,3)\n",
    "a2"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "colab_type": "text",
    "id": "54wb4NboLKxN"
   },
   "source": [
    "#### Create a 3x3 identity matrix"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 23,
   "metadata": {
    "colab": {},
    "colab_type": "code",
    "id": "j8JQg-VSLKxT",
    "outputId": "697e3eed-3266-4f57-90ab-ff5d5c430c7d"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "array([[1., 0., 0.],\n",
       "       [0., 1., 0.],\n",
       "       [0., 0., 1.]])"
      ]
     },
```

```
     "execution_count": 23,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "np.eye(3)"
   ]
  },
  {
   "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": 27,
   "metadata": {
    "colab": {},
    "colab_type": "code",
    "id": "Co4I1qACLKxc",
    "outputId": "efdd03f8-868e-4eeb-e40f-e75dba4cd14e"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "array([[0.68761205]])"
      ]
     },
     "execution_count": 27,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "x=np.random.rand(1,1)\n",
    "x"
   ]
  },
  {
   "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": 28,
   "metadata": {
    "colab": {},
    "colab_type": "code",
    "id": "eUgQBgpQLKxs",
    "outputId": "04664997-3ff0-421e-b92e-5e12c80321f9"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "array([ 0.40676241,  0.01919772, -0.53207531, -0.36309496,  0.11697544,\n",
       "        0.88428258,  0.56316995, -0.11828199, -0.96252082, -1.63962687,\n",
       "       -1.1690078 ,  0.64177579, -1.27255107,  0.20594047, -0.32968782,\n",
       "       -0.77043214, -0.6391797 ,  0.14984774, -1.16161564,  0.09739328,\n",
       "        3.08080732, -0.18831226,  0.99138594,  1.64733949,  0.5664771 ])"
      ]
     },
     "execution_count": 28,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "x=np.random.normal(0,1,25)\n",
    "x"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "colab_type": "text",
    "id": "-sL9HYlWLKx1"
   },
   "source": [
    "#### Create the following matrix:"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 32,
   "metadata": {
    "colab": {},
    "colab_type": "code",
    "id": "p2pm5Mm6LKx5",
    "outputId": "aa76f1ed-5967-4d3d-8061-72f82595c599"
   },
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "[0.   0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1  0.11 0.12 0.13\n",
      " 0.14 0.15 0.16 0.17 0.18 0.19 0.2  0.21 0.22 0.23 0.24 0.25 0.26 0.27\n",
```

```
      " 0.28 0.29 0.3  0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.4  0.41\n",
      " 0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49 0.5  0.51 0.52 0.53 0.54 0.55\n",
      " 0.56 0.57 0.58 0.59 0.6  0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69\n",
      " 0.7  0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8  0.81 0.82 0.83\n",
      " 0.84 0.85 0.86 0.87 0.88 0.89 0.9  0.91 0.92 0.93 0.94 0.95 0.96 0.97\n",
      " 0.98 0.99 1.  ]\n",
      "[0.   0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1  0.11 0.12 0.13\n",
      " 0.14 0.15 0.16 0.17 0.18 0.19 0.2  0.21 0.22 0.23 0.24 0.25 0.26 0.27\n",
      " 0.28 0.29 0.3  0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.4  0.41\n",
      " 0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49 0.5  0.51 0.52 0.53 0.54 0.55\n",
      " 0.56 0.57 0.58 0.59 0.6  0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69\n",
      " 0.7  0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8  0.81 0.82 0.83\n",
      " 0.84 0.85 0.86 0.87 0.88 0.89 0.9  0.91 0.92 0.93 0.94 0.95 0.96 0.97\n",
      " 0.98 0.99 1.  ]\n"
     ]
    }
   ],
   "source": [
    "x=np.arange(0,1.01,0.01)\n",
    "print(x)\n",
    "y=np.linspace(0,1,101)\n",
    "print(y)\n"
   ]
  },
  {
   "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": 33,
   "metadata": {
    "colab": {},
    "colab_type": "code",
    "id": "AniTWWz9LKyC",
    "outputId": "dfb092f8-2f1e-446a-99e2-5c6df71631ac"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "array([0.        , 0.05263158, 0.10526316, 0.15789474, 0.21052632,\n",
       "       0.26315789, 0.31578947, 0.36842105, 0.42105263, 0.47368421,\n",
       "       0.52631579, 0.57894737, 0.63157895, 0.68421053, 0.73684211,\n",
       "       0.78947368, 0.84210526, 0.89473684, 0.94736842, 1.        ])"
      ]
     },
     "execution_count": 33,
     "metadata": {},
     "output_type": "execute_result"
```

```
    }
   ],
   "source": [
    "np.linspace(0,1,20)"
   ]
  },
  {
   "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": null,
   "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": 38,
     "metadata": {
      "tags": []
     },
     "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": null,
   "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": 40,
     "metadata": {
      "tags": []
     },
     "output_type": "execute_result"
    }
   ],
   "source": [
    "mat[2:5,1:5]"
   ]
  },
  {
   "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": null,
   "metadata": {
    "colab": {},
    "colab_type": "code",
    "id": "QJyclCVfLKy6",
    "outputId": "9762e12a-13fe-4567-fd7a-718402be1edb"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "20"
      ]
     },
     "execution_count": 41,
     "metadata": {
      "tags": []
     },
     "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": null,
   "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": 42,
     "metadata": {
      "tags": []
     },
     "output_type": "execute_result"
    }
   ],
   "source": [
    "mat[0:3,1:2]"
   ]
  },
  {
   "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": null,
   "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": 46,
     "metadata": {
      "tags": []
     },
     "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": null,
   "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": 49,
     "metadata": {
      "tags": []
     },
     "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": null,
   "metadata": {
    "colab": {},
    "colab_type": "code",
    "id": "lgjWcsNHLKz9",
    "outputId": "a4aebb34-800e-490a-94fa-b2f29e05ec8a"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "325"
      ]
     },
     "execution_count": 50,
     "metadata": {
      "tags": []
     },
     "output_type": "execute_result"
    }
   ],
   "source": [
    "mat.sum()"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "colab_type": "text",
    "id": "aOxAVLe8LK0B"
   },
   "source": [
    "#### Get the standard deviation of the values in mat"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "colab": {},
    "colab_type": "code",
    "id": "1bExi0tvLK0D",
    "outputId": "cdcf114d-bcd3-494a-afc8-f4b592566334"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
```

```
        "7.2111025509279782"
       ]
      },
      "execution_count": 51,
      "metadata": {
       "tags": []
      },
      "output_type": "execute_result"
     }
    ],
    "source": [
     "mat.std()"
    ]
   },
   {
    "cell_type": "markdown",
    "metadata": {
     "colab_type": "text",
     "id": "qWvkrLQWLK0H"
    },
    "source": [
     "#### Get the sum of all the columns in mat"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
     "colab": {},
     "colab_type": "code",
     "id": "-XFw4SD1LK0J",
     "outputId": "03cee652-bdb1-411c-a81f-7534525a3d21"
    },
    "outputs": [
     {
      "data": {
       "text/plain": [
        "array([55, 60, 65, 70, 75])"
       ]
      },
      "execution_count": 53,
      "metadata": {
       "tags": []
      },
      "output_type": "execute_result"
     }
    ],
    "source": [
     "sum(mat)"
    ]
   },
   {
    "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": "3.8.0"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 1
}
```