جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

College of Science
Department of Intelligent Medical Systems

Lecture_Lab#3

Prof. Dr. Mehdi Ebady Manaa

دالة حساب كتلة الجسم

```
def calculate_bmi(weight_kg, height_m):
    """Calculate Body Mass Index (BMI)."""
    if height_m <= 0:
        return "Invalid height"
    return round(weight_kg / (height_m ** 2), 2)
# Example usage
bmi = calculate_bmi(70, 1.75)
print("BMI:", bmi)
```

شرح الدالة `calculate_bmi(weight_kg, height_m)` في بايثون

الدالة `calculate_bmi` تُستخدم لحساب **مؤشر كتلة الجسم (BMI - Body Mass Index)** بناءً على **الوزن بالكيلوغرام** و**الطول بالمتر**.

حساب نطاق معدلات ضربات القلب حسب معادلة كارنوفين

```
def heart_rate_zones(age, resting_hr):
    """Calculate heart rate training zones based on Karvonen Formula."""
    max_hr = 220 - age
    reserve_hr = max_hr - resting_hr
    zones = {
        "Light": (resting_hr + reserve_hr * 0.5, resting_hr + reserve_hr * 0.6),
        "Moderate": (resting_hr + reserve_hr * 0.6, resting_hr + reserve_hr * 0.7),
        "Hard": (resting_hr + reserve_hr * 0.7, resting_hr + reserve_hr * 0.8),
        "Very Hard": (resting_hr + reserve_hr * 0.8, resting_hr + reserve_hr * 0.9)
```

```
    }
    return {zone: (round(low), round(high)) for zone, (low, high) in zones.items()}
# Example usage
zones = heart_rate_zones(30, 60)
print("Heart Rate Zones:", zones)
```

شرح المعادلة:

تستخدم **معادلة كارفونين (Karvonen Formula)** لحساب **معدل ضربات القلب المستهدف (THR)** بناءً على نطاقات تدريبية مختلفة لمساعدتك في تحسين التمارين الرياضية.

حيث:

- **احتياطي معدل ضربات القلب = (HRR) الحد الأقصى لمعدل ضربات القلب - معدل ضربات القلب أثناء الراحة**
- **الحد الأقصى لمعدل ضربات القلب - 220 = (Max HR) العمر**
- تحدد **نطاقات التمارين** وفقًا لمستويات الشدة المختلفة.

حساب كمية استهلاك السعرات الحرارية

```
def calories_burned(weight_kg, duration_min, met):
    """Calculate calories burned during an activity.
    MET (Metabolic Equivalent of Task) values are used."""
    return round((met * weight_kg * 3.5 / 200) * duration min, 2)

# Example usage (Running at 8 km/h ~ MET 8.0)
calories = calories_burned(70, 30, 8) x
print("Calories Burned:", calories)
```

شرح دالة حساب كمية استهلاك السعرات الحرارية `calories_burned`

الدالة `calories_burned(weight_kg, duration_min, met)` تُستخدم **لحساب عدد السعرات الحرارية التي يحرقها الشخص أثناء ممارسة نشاط معين**، بناءً على **وزنه ومدّة التمرين ومستوى النشاط البدني.(MET)**

📌المعادلة الرياضية لحساب السعرات الحرارية المحروقة

📌 المعادلة الرياضية لحساب السعرات الحرارية المحروقة

تُحسب السعرات الحرارية المحروقة باستخدام **معادلة MET**:

$$\text{السعرات المحروقة} = \left(\frac{\text{MET} \times \text{وزن الجسم بالكغم} \times 3.5}{200} \right) \times \text{مدة التمرين بالدقائق}$$

🔹 مكونات المعادلة:

- ✅ `weight_kg` **(الوزن بالكيلوغرام)** → كلما زاد الوزن، زاد حرق السعرات الحرارية.
- ✅ `duration_min` **(مدة التمرين بالدقائق)** → مدة النشاط البدني.
- ✅ `met` **(معدل MET - Metabolic Equivalent of Task)** → مقياس يعبر عن **معدل استهلاك الطاقة** لنشاط معين مقارنةً بحالة الراحة.

دالة حساب ضغط الدم

```
def classify_blood_pressure(systolic, diastolic):
    """Classify blood pressure based on American Heart Association guidelines."""
    if systolic < 120 and diastolic < 80:
        return "Normal"
    elif 120 <= systolic < 130 and diastolic < 80:
        return "Elevated"
    elif 130 <= systolic < 140 or 80 <= diastolic < 90:
        return "Hypertension Stage 1"
    elif systolic >= 140 or diastolic >= 90:
        return "Hypertension Stage 2"
    else:
        return "Hypertensive Crisis - Seek medical attention!"

# Example usage
bp_status = classify_blood_pressure(135, 85)
print("Blood Pressure Category:", bp_status)
```

🔍شرح دالة حساب ضغط الدم `classify_blood_pressure(systolic, diastolic)`

الدالة `classify_blood_pressure`تُستخدم لتحديد **تصنيف ضغط الدم** بناءً على **الضغط الانقباضي (Systolic)** **والضغط الانبساطي (Diastolic)** وفقًا للمعايير الطبية.

📌ما هو ضغط الدم؟

ضغط الدم هو **القوة التي يضغط بها الدم على جدران الشرايين أثناء ضخ القلب للدم**.

- **الضغط الانقباضي (Systolic):** هو الرقم العلوي، ويمثل **الضغط أثناء انقباض القلب وضخ الدم**.
- **الضغط الانبساطي (Diastolic):** هو الرقم السفلي، ويمثل **الضغط عندما يكون القلب في حالة راحة بين النبضات**.

📌 كيف يتم تصنيف ضغط الدم؟

وفقًا لجمعية القلب الأمريكية (AHA)، يتم تصنيف ضغط الدم كالتالي:

التصنيف	الضغط الانقباضي (Systolic)	الضغط الانبساطي (Diastolic)
🔵 طبيعي	أقل من 120	أقل من 80
🟡 مرتفع قليلاً	129 - 120	أقل من 80
🟠 ارتفاع ضغط الدم (المرحلة 1)	139 - 130	89 - 80
🔴 ارتفاع ضغط الدم (المرحلة 2)	140 أو أكثر	90 أو أكثر
🚨 أزمة ارتفاع الضغط	أكثر من 180	أكثر من 120

دالة **حويل عدد الخطوات إلى المسافة بالمتر** يعني **حساب المسافة التي يقطعها الشخص بناءً على عدد الخطوات التي يمشيها أو يركضها.**

```
def steps_to_distance(steps, stride_length_m):
    """Convert steps to distance in meters."""
    return round(steps * stride_length_m, 2)

# Example usage (assuming average stride length is 0.78 meters)
distance = steps_to_distance(10000, 0.78)
print("Distance Walked (m):", distance)
```

📌 كيف يتم حساب المسافة؟

تُستخدم المعادلة التالية لتحويل عدد الخطوات إلى المسافة:

$$\text{المسافة} = \text{عدد الخطوات} \times \text{طول الخطوة}$$

✅ **عدد الخطوات (Steps):** العدد الإجمالي للخطوات التي يمشيها أو يركضها الشخص.
✅ **طول الخطوة (Stride Length):** الطول المتوسط للخطوة **بالمتر**، والذي يعتمد على طول الشخص ونمط المشي أو الجري.

- المشي: عادةً **0.7 - 0.8 متر** للبالغين.
- الجري: عادةً **1.0 - 1.2 متر** أو أكثر.

دالة **تقدير درجة جودة النوم** يعني **تقييم جودة النوم** بناءً على **عدد ساعات النوم، عدد مرات الاستيقاظ أثناء النوم، ونسبة النوم العميق**.

```
def sleep_quality(hours_slept, wakeups, deep_sleep_percent):
    """Estimate sleep quality score (0-100)."""
    score = (hours_slept * 10) - (wakeups * 5) + (deep_sleep_percent * 0.5)
    return max(0, min(100, round(score)))

# Example usage
sleep_score = sleep_quality(7.5, 2, 50)
print("Sleep Quality Score:", sleep_score)
```

📌 كيف يتم حساب جودة النوم؟

تُحسب **درجة جودة النوم (0-100)** باستخدام المعادلة التالية:

$$\text{درجة جودة النوم} = (\text{عدد ساعات النوم} \times 10) - (\text{عدد مرات الاستيقاظ} \times 5) + (\text{نسبة النوم العميق} \times 0.5)$$

✅ **كل ساعة نوم تضيف 10 نقاط**
✅ **كل استيقاظ ينقص 5 نقاط**
✅ **كل 1% من النوم العميق يضيف 0.5 نقطة**
✅ **النسبة النهائية تُقيد بين 0 و 100 (لا يمكن أن تكون أقل من 0 أو أكثر من 100).**

دالة **تقدير كمية الماء اليومية (باللترات) بناءً على الوزن ومستوى النشاط** يعني **حساب كمية الماء التي يحتاجها الجسم يوميًا** اعتمادًا على **وزن الشخص ومستوى نشاطه البدني**.

```
def daily_water_intake(weight_kg, activity_level):
    """Estimate daily water intake (liters) based on weight and activity level."""
    base_intake = weight_kg * 0.033  # Base intake in liters
    extra_intake = activity_level * 0.5  # Extra liters for each hour of exercise
    return round(base_intake + extra_intake, 2)
```

```
# Example usage
water_needed = daily_water_intake(70, 1)  # 1 hour of activity
print("Water Intake (L):", water_needed)
```

📌 كيف يتم حساب كمية الماء اليومية؟

يحتاج الجسم إلى كمية معينة من الماء يوميًا للحفاظ على **الترطيب السليم، وظائف الأعضاء، وتنظيم درجة الحرارة**.
يتم حساب هذه الكمية باستخدام المعادلة التالية:

$$\text{كمية الماء اليومية (لتر)} = \text{الوزن (كغم)} \times 0.033 + (\text{عدد ساعات النشاط البدني} \times 0.5)$$

✅ **الوزن** (`weight_kg`) → كلما زاد وزن الشخص، زادت حاجته إلى الماء.
✅ **مستوى النشاط البدني** (`activity_level`) → كلما كان الشخص أكثر نشاطًا، زادت حاجته للماء بسبب التعرق وفقدان السوائل.

حساب دالة حالة السكر

```
def glucose_status(fasting_glucose):
    """Classify blood glucose levels based on fasting glucose."""
    if fasting_glucose < 70:
        return "Low (Hypoglycemia)"
    elif 70 <= fasting_glucose < 100:
        return "Normal"
    elif 100 <= fasting_glucose < 126:
        return "Prediabetes"
    else:
        return "Diabetes - Seek medical advice"
# Example usage
glucose_category = glucose_status(110)
print("Glucose Level Status:", glucose_category)
```

📌 ما هو تحليل السكر الصائم؟

تحليل السكر الصائم (Fasting Blood Glucose - FBG) هو اختبار يُستخدم لتقييم مستويات **السكر في الدم** وتشخيص مرض **السكري** أو **مرحلة ما قبل السكري**.

📊 تصنيف مستويات السكر وفقًا لمنظمة الصحة العالمية (WHO) والجمعية الأمريكية للسكري (ADA):

التصنيف	مستوى السكر الصائم (mg/dL)
🔵 طبيعي	أقل من 100 mg/dL
🟡 مرحلة ما قبل السكري	125 - 100 mg/dL
🔴 السكري	126 mg/dL أو أكثر

📌 **ملاحظة:**

- إذا كانت القراءة **126 mg/dL أو أكثر في اختبارين منفصلين**، فهذا يُشير إلى **الإصابة بمرض السكري**.
- الأشخاص في **مرحلة ما قبل السكري** يجب أن يكونوا أكثر حرصًا على **تعديل نمط حياتهم** لمنع الإصابة بالسكري.

```
def cholesterol_risk(total_cholesterol, hdl, ldl):
    """Classify cholesterol levels."""
    if total_cholesterol < 200 and hdl >= 60 and ldl < 100:
        return "Optimal"
    elif total_cholesterol >= 240 or ldl >= 160:
        return "High Risk"
    elif 200 <= total_cholesterol < 240 or ldl >= 130:
        return "Borderline High"
    else:
        return "Moderate Risk"

# Example usage
cholesterol_status = cholesterol_risk(210, 55, 140)
print("Cholesterol Risk Category:", cholesterol_status)
```

📌 ما هو الكوليسترول؟

الكوليسترول هو **مادة دهنية ضرورية للجسم**، لكن ارتفاعه بشكل غير طبيعي قد يسبب **مشاكل في القلب والشرايين**.

الكوليسترول الكلي (`Total Cholesterol`) = مجموع الكوليسترول الجيد والضار والدهون الثلاثية.

الكوليسترول الجيد (`HDL`) → **جيد للقلب** لأنه يساعد في إزالة الكوليسترول الزائد.

الكوليسترول الضار (`LDL`) → **ضار للقلب** لأنه يسبب تراكم الدهون في الشرايين.

📌 تصنيف مستويات الكوليسترول وفقًا لمنظمة الصحة العالمية (WHO)

التصنيف	الكوليسترول الكلي (mg/dL)	HDL (mg/dL)	LDL (mg/dL)
🔵 طبيعي	أقل من 200	أكثر من 60	أقل من 100
🟡 حدود مرتفعة	239 - 200	59 - 40	159 - 100
🔴 خطر مرتفع	240 أو أكثر	أقل من 40	160 أو أكثر

ملحوظة:

- كلما زاد HDL (الكوليسترول الجيد)، كان ذلك أفضل للصحة.
- كلما زاد LDL (الكوليسترول الضار)، زاد خطر الإصابة بأمراض القلب.

```
def stress_level(heart_rate, sleep_hours, daily_exercise_minutes):
    """Estimate stress level based on heart rate, sleep, and exercise."""
    score = (heart_rate / 2) - (sleep_hours * 3) - (daily_exercise_minutes * 0.5)
    if score < 20:
        return "Low Stress"
    elif 20 <= score < 40:
        return "Moderate Stress"
    else:
        return "High Stress - Consider relaxation techniques"

# Example usage
stress = stress_level(80, 6, 30)
print("Stress Level:", stress)
```

📌 كيف يتم حساب مستوى التوتر؟

يتم حساب درجة التوتر (Score) باستخدام المعادلة التالية:

$$\text{درجة التوتر} = \left(\frac{\text{معدل ضربات القلب}}{2}\right) - (\text{عدد ساعات النوم} \times 3) - (\text{مدة التمارين بالدقائق} \times 0.5)$$

🔹 تفسير المعادلة:

✅ زيادة معدل ضربات القلب → زيادة مستوى التوتر
✅ زيادة عدد ساعات النوم → تقليل التوتر
✅ زيادة مدة التمارين الرياضية → تقليل التوتر